



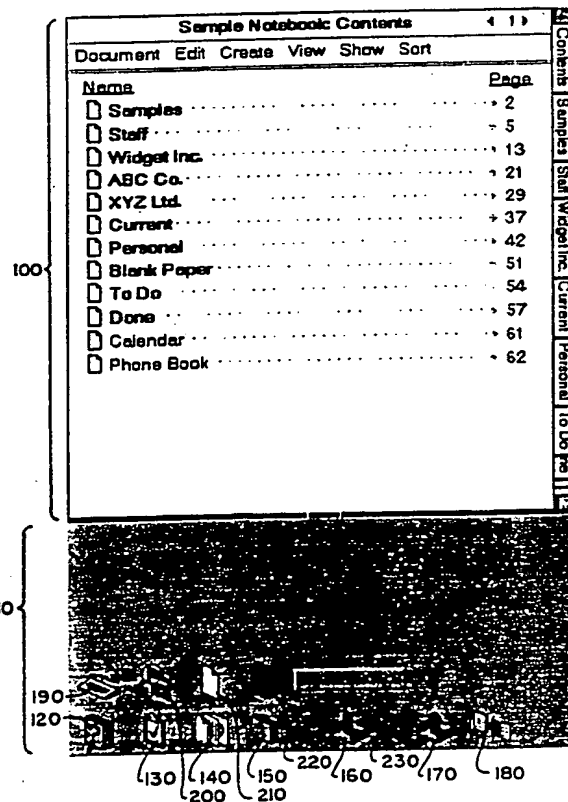
INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) International Patent Classification ⁵ : G06F 15/40, 9/44, 3/033 G06F 3/023		A1	(11) International Publication Number: WO 92/08199 (43) International Publication Date: 14 May 1992 (14.05.92)
(21) International Application Number: PCT/US91/08077 (22) International Filing Date: 29 October 1991 (29.10.91) (30) Priority data: 607,139 31 October 1990 (31.10.90) US (60) Parent Application or Grant (63) Related by Continuation US 607,139 (CIP) Filed on 31 October 1990 (31.10.90) (71) Applicant (for all designated States except US): GO CORPORATION [US/US]; 950 Tower Lane, Suite 1400, Foster City, CA 94404 (US). (72) Inventors; and (75) Inventors/Applicants (for US only) : CARR, Robert [US/US]; 725 Euclid Ave., San Francisco, CA 94118 (US). FRAISL, Daniel, J. [US/US]; 21697 Lomita Ave., Cupertino, CA 95014 (US). ISAAC, Steve [US/US]; 12 East Court Lane, Foster City, CA 94404 (US). OUYE, Michael [US/US]; 840 Boyce Ave., Palo Alto, CA 94301 (US).		(74) Agent: RADLO, Edward, J.; Fenwick & West, Two Palo Alto Square, Palo Alto, CA 94306 (US). (81) Designated States: AT, AT (European patent), AU, BB, BE (European patent), BF (OAPI patent), BG, BJ (OAPI patent), BR, CA, CF (OAPI patent), CG (OAPI patent), CH, CH (European patent), CI (OAPI patent), CM (OAPI patent), DE, DE (European patent), DK, DK (European patent), ES, ES (European patent), FI, FR (European patent), GA (OAPI patent), GB, GB (European patent), GN (OAPI patent), GR (European patent), HU, IT (European patent), JP, KP, KR, LK, LU, LU (European patent), MC, MG, ML (OAPI patent), MR (OAPI patent), MW, NL, NL (European patent), NO, RO, SD, SE, SE (European patent), SN (OAPI patent), SU*, TD (OAPI patent), TG (OAPI patent), US.	
		Published With international search report. Before the expiration of the time limit for amending the claims and to be republished in the event of the receipt of amendments.	

(54) Title: COMPUTER DOCUMENTS AS COMPOUND DOCUMENTS IN A NOTEBOOK METAPHOR

(57) Abstract

A method of implementing a notebook metaphor for a computer, in which all of the documents stored in the computer are embedded within a single document. The computer automatically places all created documents in a table of contents (10), and these documents are accessible by tabs (450), page numbers (420), and the like. Documents (1490) may be embedded within one another in a fashion which allows full access to each of the various applications used in creating the documents. These applications are automatically accessed, and are run simultaneously in a multi-tasking manner. As a whole, the system imitates, in an actively stored computer memory with automatic updating, a notebook (10) with a virtual bookshelf (110) for accessing miscellaneous applications, and with indexing (1270) and tabs (450) for accessing different documents.



+ DESIGNATIONS OF "SU"

Any designation of "SU" has effect in the Russian Federation. It is not yet known whether any such designation has effect in other States of the former Soviet Union.

FOR THE PURPOSES OF INFORMATION ONLY

Codes used to identify States party to the PCT on the front pages of pamphlets publishing international applications under the PCT.

AT	Austria	ES	Spain	MG	Madagascar
AU	Australia	FI	Finland	ML	Mali
BB	Barbados	FR	France	MN	Mongolia
BE	Belgium	GA	Gabon	MR	Mauritania
BF	Burkina Faso	GB	United Kingdom	MW	Malawi
BG	Bulgaria	GN	Guinea	NL	Netherlands
BJ	Benin	GR	Greece	NO	Norway
BR	Brazil	HU	Hungary	PL	Poland
CA	Canada	IT	Italy	RO	Romania
CF	Central African Republic	JP	Japan	SD	Sudan
CG	Congo	KP	Democratic People's Republic of Korea	SE	Sweden
CH	Switzerland			SN	Senegal
CI	Côte d'Ivoire	KR	Republic of Korea	SU ⁺	Soviet Union
CM	Cameroon	LI	Liechtenstein	TD	Chad
CS	Czechoslovakia	LK	Sri Lanka	TC	Togo
DE	Germany	LU	Luxembourg	US	United States of America
DK	Denmark	MC	Monaco		

COMPUTER DOCUMENTS AS COMPOUND
DOCUMENTS IN A NOTEBOOK METAPHOR

Technical Field

5 The present invention is directed towards the problem of constructing, organizing, accessing and linking documents on a computer. Such documents may contain text, graphic images, geometric figures, or numerical spreadsheets, or may be compound documents made up of nested or embedded sub-documents.

10

Background Art

There are many inventions addressing this area in the prior art. Most are based on the idea of a "Graphical User Interface" which attempts to model some part of the real world on a computer screen. Previous work commonly
15 uses a "desk-top metaphor" in which the computer's display is treated as a desk-top upon which objects such as documents or computer programs are represented as symbols or icons. These icons may be "dragged" to any location on the desk-top by using a pointing device such as a mouse. Desk-top objects may be aggregated by collection into any number of folders. These
20 folders may be placed inside of other folders. The Xerox Star computer and the Macintosh from Apple Computer are examples of such work. Much of this prior work uses an "application program & data file" view of the world in which the user is aware that an application program is executing which can operate on a number of data files or documents.

25

Information in the area of software related to this invention can be found in Inside Macintosh, Fundamentals of Interactive Computer Graphics, SmallTalk-80, The Language and Its Implementation (all from Addison-Wesley Publishing Co.), "The X Window System" (ACM Transactions on Graphics, Vol. 5, No. 2, April 1986, pages 79-109) or "Andrew: A Distributed
30 Personal Computing Environment" (Communications of the ACM, March 1986). The actual techniques use to recognize both gestures and handwritten input will not be discussed in detail here, but may be found in "Automatic

Recognition of Handprinted Characters - The State of the Art", Proceedings of the IEEE, pages 469-487, Vol. 68, No. 4, April 1980. The foregoing references are incorporated herein by reference.

5 As organizing frameworks, desk-tops and folders leave much to be desired. Objects within these "containers" can be arranged in any pattern with no structuring supported within the container, except further containment or nesting. Indeed, many previous systems support commands to "Clean up the window" or "Arrange icons". Often objects may be displayed as a list of titles sorted by name, size, date, or type but the user has no way to
10 impose his arbitrary order on the list.

The "program & file" model intrudes in a number of ways. First: such a design requires a load/save metaphor, in which the user must tell the program to load or save a data file. Second: when one program is operating upon many files, the user must select which file the program's attention
15 should be focused on. This is because it must be clear what is the target of shared controls such as command menus.

A number of different approaches to the construction of compound documents exist in prior work. One approach is to simply merge a static image of the foreign document into the parent file. Such an image is "dead",
20 in that it can no longer be manipulated in its native format. Thus, a text document could contain an embedded pie chart, but the ability to vary the underlying quantities that produced the chart would be lost. These static images are commonly stored as "bit-maps", or arrays of pixels. Bit-maps are difficult to re-size because the underlying structure has been lost.

25 Another scheme is to have the parent document's controlling program recognize the imported file's format. For full generality, this requires that all application programs understand all data file formats.

A third approach is to simply provide a window in the parent document in which the program controlling the sub-document displays its
30 output. In order for both the parent and child documents to be responsive to manipulation, both application programs must be executing. This implies some form of multi-tasking operating system. Such an approach is taken by

the current invention.

The prior art contains certain examples of application programs which, on one level, exhibit some of the behavior of a notebook. "Hypercard" from Apple Computer and "Current" from IBM both can present an image of a notebook with page-turn buttons, tabs and an index. The tabs and index entries can be selected with the mouse cursor to turn to the corresponding pages.

A major difference between these existing application programs and the current invention is that their notebooks are static objects contained within a single application program. The number of pages, page contents, buttons, tabs, and index entries were laid out by the designer of the application, and do not change in normal operation. In contrast, the notebook in the present invention is designed to control an indefinite number of documents created by a variety of diverse application programs such as word processors, numeric spreadsheets, drawing programs, address lists, FAX viewers, and computerized forms. A notebook contains an arbitrary and changeable number of these documents, one per page, which need have no knowledge that they are embedded in a notebook. The notebook in the present invention acts as the "job control language" for a collection of programs and their associated files. The contents of the table of contents and index change as documents are created, moved or deleted. The visible bookmarks or tabs can be enabled on a document-by-document basis. In addition, any number of pages in the present invention can be "removed" from their home position and "floated" over another page and then sent home. No known work in the prior art exhibits this set of features. Or, to state it another way, although the present invention could be used to build something much like "Hypercard", the converse is definitely not true.

Summary of the Invention

The present invention includes methods and apparatus for controlling a computer. The computer presents the behavior of a multi-page notebook that resides on a bookshelf containing other notebooks and accessories.

Features include visible numbered pages, page-turn effects, tabbed bookmarks along one or more edge, "floating" pages temporarily removed from their notebook location, a table of contents, contents of sub-sections, and an index. A "Document Model" is presented to the user in which the existence of an
5 underlying computer program implementing the behavior of a document is hidden.

Each page in the notebook contains one document which may contain arbitrary amounts of data. These pages are analogous to the combination of data files and application programs in other computers, and may be
10 compound documents containing different types of documents such as text, drawings, FAX images, or specialized forms.

The user has complete control over the ordering of his documents by placing them where he wishes in the notebook. In addition, arbitrary linkages may be made between documents by easily-created "link buttons"
15 placed in one document and tied to an object in another. When such a button is selected, the page of its target is turned to. The construction of compound documents is supported and implemented by the computer's operating system, and does not require an application program to contain any special code to support sub-documents or to be a sub-document.

20 The notebook metaphor presents a virtual object which everyone is familiar with and can use: a book. Components such as the table of contents, page numbers, and page tabs are instantly recognizable and can be manipulated by a novice user. The match to a real-world object is even stronger when this metaphor is implemented on a "note pad"-style computer
25 which uses a stylus as a pointing device directly on the screen.

Such a computer is described in detail in another United States patent application filed by applicants herein, namely Agulnick et al., "Control of a Computer Through a Position-Sensed Stylus," Serial No. 07/610,231, filed October 31, 1990. That application, including its appendices, are incorporated
30 herein by reference. The appendices are all from GO Corporation of Foster City, California, and are entitled:

I. Architecture—Parts 0 to 5

II. Architecture--Parts 6 to End

III. Application Developer's Guide

IV. API Reference.

Another copending application which complements the disclosure
5 herein is the United States patent application of Hullender entitled "Method
For Pattern Recognition," Serial No. 07/607,125, filed on October 31, 1990.
That application is also incorporated herein by reference.

The notebook of the present invention allows the user to impose a
variety of structures on his documents. The structure can be as simple as a
10 list, or divided into sections, or even a large hierarchy with nested sections.
This structure itself can help the user remember where documents are, and
can place related documents together, and also supports a variety of retrieval
paths. The user can flip from page to page, or turn to a specific page by
selecting its tab or entry in a table of contents or index. The table of contents
15 and index provide complementary browsing mechanisms, one ordered by
page and one by name.

Because the functions of embedding and the construction of compound
documents and link buttons are supported across-the-board in the operation
system, these functions are available in all application programs. In addition,
20 the operating system is free to use these functions to build its own user
interface.

The entries in the table of contents and index along with page tabs are
all forms of link buttons. The bookshelf, notebooks, table of contents and
documents are all independent programs embedded one within the other.
25 This support for embedding is built of robust, well isolated operating system-
level objects such as independent tasks, file directories, and display windows.

The present invention implements an Operating System and a
Graphical User Interface (GUI) on a computer's graphical display. In a GUI,
the computer presents a simplified simulation of some portion of the real
30 world. Objects within the computer such as files, directories and programs
are represent on the display as small pictures or icons. These icons are
selected or moved around the screen by using some pointing device such as a

mouse or electronic stylus. The goal is to provide the user with an intuitive feel for the behavior of the objects stored by the computer. This is done by allowing for their direct manipulation with the pointing device. Objects may be "opened", "closed", or "dragged" onto/into other objects.

5 The most common pointing device is a mouse with one to three buttons. A cursor on the screen tracks the movement of the mouse. Commands are issued by moving the cursor over some area of the screen and pressing one of the mouse buttons. Pressing and releasing a button is a "click" which can be extended to double and triple clicking. Holding a button
10 down and moving the mouse generally is used to drag the object beneath the cursor.

 An electronic stylus whose position is sensed by a digitizer can also be used as a pointing device. In addition to the actions available with a mouse, the stylus adds gestures. For example an object can be deleted by drawing an
15 'X' over it. If the digitizer is integrated with the display, then the cursor may be dispensed with, because the stylus can point directly to displayed objects.

 The present invention can thus be used with a mouse, stylus or other pointing devices.

 The implementation of the actual "notebook" GUI software can be
20 based on any one of many prior inventions. Information in this area can be found in *Inside Macintosh*, *Fundamentals of Interactive Computer Graphics*, *SmallTalk-80*, *The Language and Its Implementation* (all from Addison-Wesley Publishing Co.), "The X Window System" (*ACM Transactions on Graphics*, Vol. 5, No. 2, April 1986, pp. 79-109) or "Andrew: A Distributed
25 Personal Computing Environment" (*Communications of the ACM*, March 1986). These references are incorporated herein by reference.

Brief Description of the Drawings

 Figure 1 shows an initial display of the notebook-metaphor computer
30 of the present invention.

 Figure 2 shows the opening of a window related to an icon on the bookshelf.

Figures 3-5 show different types of icons available to the user.

Figure 6 shows the re-sizing of the displayed main notebook.

Figure 7 shows a document comprising a drawing.

Figure 8 shows a document comprising text.

5 Figure 9 shows the document of Figure 8, scrolled down to a lower portion of the document.

Figure 10 shows a document utilizing various features common to all documents in the notebook-metaphor system.

10 Figure 11 shows the document of Figure 10, with the document menu displayed.

Figure 12 shows the document of Figure 10, with an option sheet floating above it.

Figure 13 shows the document of Figure 12, with the displayed list of accessible sheets.

15 Figure 14 shows the document of Figure 13, with the access sheet floating above it.

Figure 15 shows a text document with an embedded drawing document.

20 Figure 16 shows the text document of Figure 15, with the border of the embedded document disabled.

Figure 16 shows compound documents built from other documents.

Figure 17 shows a text options sheet floated above a document.

Figure 18 shows the document of Figure 17, with an Edit menu floated above it.

25 Figure 19 shows a Table of Contents with a help notebook floated above it.

Figure 20 shows a contents notebook.

Figure 21 shows the contents notebook with the View menu displayed.

30 Figure 22 shows the contents notebook of Figure 21, with one of the entries expanded.

Figure 23 shows the contents notebook of Figure 21, with all subsections expanded.

Figure 24 shows a Table of Contents with the Show menu displayed.

Figure 25 shows the Table of Contents of Figure 24, with the hidden page tabs displayed.

Figure 26 shows a contents page with a FAX page floating above it.

5 Figure 27 shows the FAX page of Figure 26, expanded to full size.

Figures 28 and 29 show the re-sizing of the FAX page of Figure 26.

Figures 30 and 31 show the repositioning of the FAX page of Figure 26.

Figure 32 shows the contents page with the Edit menu displayed.

Figure 33 shows the contents page with the Create menu displayed.

10 Figure 34 shows the contents page with the Sort menu displayed.

Figure 35 shows the index page of the notebook.

Figure 36 shows a document with numerous link buttons linked to headings in the document.

15 Figure 37 shows a document with a link button embedded in the document.

Figure 38 shows a link button from one document embedded into another document.

Figure 39 shows the floating of one document above another due to double-clicking of a link button.

20 Figure 40 shows a link button linked to a word in the help document.

Figure 41 shows the manner of storage of documents in the file system of the invention.

Figure 42 shows a compound document with embedded documents.

Figure 43 shows the editing of an embedded document.

25 Figure 44 shows the result of the editing shown in Figure 43.

Figure 45 shows the in-line emplacement of an embedded text document.

Figure 46 shows an embedded drawing document with its borders displayed.

30 Figure 47 shows the document of Figure 47, with the Access menu displayed.

Figure 48 shows the result of removing the borders from the embedded

document in Figures 46 and 47.

Description of the Preferred Embodiments

5 Figure 1 shows a typical initial display or screen of the notebook computer. The upper portion of the display shows the "table of contents" display 100 of the main notebook 10. The lower portion 110 is a "bookshelf" of icons which represent notebooks 120, various accessories 200 or any other object that the user wishes to place there, such as a link button 230.

10 The bookshelf is the highest level "document" in the system, and all other objects or their containing documents are found there. Clicking or tapping (as with a stylus) on an icon 150 on the bookshelf will open a window 240 for that item. When an item is open, its icon is presented as a gray shadow 250 of the normal icon. The main notebook resides on the bookshelf but is open, and therefore its icon 260 presents the shadow effect. The objects
15 (such as icons and buttons) embedded within the bookshelf are available to the user of the computer from any of the pages stored by the system.

Some of the other items shown on the bookshelf 110 are: the "Help Information" notebook 120; the document containing users preferences 130; the stationery notebook 140, the accessories container 150, the in and out
20 boxes 160 and 170, respectively; the application program installer 180; the "on screen" keyboard 190; the disk browser 200; and the installed devices notebook 210. The user can set a preference controlling the format of the icons on the bookshelf. The possible formats are large icons as shown in Figure 1, large icons with titles 270 (see Figure 3), small icons 280 (Figure 4), or small icons
25 with titles 290 (Figure 5).

The main notebook 10 floats over the bookshelf. As shown in Figure 6, the lower edge 300 of the main notebook 10 can be dragged up or down to reveal more or less of the bookshelf 110. This re-sizing is done by dragging the box 310 at the center vertically with the pointing device.

30 A notebook is composed of a series of pages numbered sequentially by the software of the invention with the first page (the table of contents) being page one. A page is a document of some type; i.e., each document is assigned

to a page by the program of the invention. The table of contents and other sub-section dividers are all types of document. Other types of documents are text (Figure 8), numerical spreadsheets, drawings (Figure 7), FAX images (Figure 27), charts, or compound documents (Figure 16) built of other documents. Thus, each document is created by at least one application, and different documents are sequentially ordered by the operating system of the invention.

Each page in a notebook is a view into some (possibly larger) document. Thus a page could represent a 30 page text document or a 6 by 9 foot drawing. For documents where the window is smaller than the full document, any portion of the document can be viewed by scrolling the document horizontally or vertically through the page window, using scroll margins 320 shown in Figure 7. A notebook can be made to mimic the behavior of a physical book by simply restricting the contents of each page to what will fit in the visible portion of the page. The Help notebook 790 (see Figure 19) is an example of this; as seen in that figure, the Help notebook is organized as a plurality of pages with a sequence and numbering which are independent of the sequence and numbering of the Sample Notebook (over which the Help notebook is shown floating).

Figure 8 shows a text document with a scroll margin 340. Documents can contain vertical and horizontal scroll margins in any combination. In general, scroll margins are made up of four elements: a slider 350, selection area 360, and the direction arrows 370 and 380.

The position of the slider 350 shows the position of the visible portion relative to the whole document. This visible page display (or document display) region on the screen can be moved by using the pointing device with the selection area or direction arrows. A click on a direction button moves the view of the window into the document one line in that direction. Tapping or clicking within the selection area causes the region of the document which corresponds to the location of the tap to be displayed in the window. The slider may be dragged up or down to a new position 390, causing the display of the corresponding region 400, as shown in Figure 9.

Figure 10 shows features such as the page title bar 410, page number 420, page turn buttons 430 and 440, and optional visible page tabs or bookmarks 450, which are features common to all pages presented in the notebook metaphor of the invention. The page names or titles, page numbers and tabs serve as page identifiers associated with the sequentially ordered plurality of pages. The page tabs are available to the user from any page, and are ordered in accordance with the sequential ordering of the pages; similarly, the tabs are automatically reordered to reflect any reordering in the sequence of pages.

As discussed below relative to embedding, a page identifier (such as "Org Chart" in Figure 25) may refer to a page which includes other pages embedded within it.

The menu bar 460 occurs on many pages in the notebook, and includes one or more buttons 470. Typically, the activation of these buttons displays sub-menus of related commands or options.

The "Document" menu 480 and "Edit" menu 490 are also available in all documents. An optional "cork margin" 500 may appear at the bottom of the page. The cork margin is essentially a private bookshelf for each document (or page) and may contain embedded documents, accessories or link buttons 510.

The Document menu 520 shown in Figure 11 provides access to commands which operate on the whole document. The "About..." button 530 causes a stack of sheets 590 (see Figure 12) to float over the current page. These "option sheets" display and control attributes of the document as a whole. The three buttons 600, 610, and 620 at the bottom of the sheet 590 are used to apply the options specified in the sheet and close the stack.

Other sheets in the stack are accessed by clicking on the current sheet's title 640 and then choosing from the displayed list 660, as shown in Figures 12 and 13. The "Document" sheet 590 provides access to such attributes as title, author, comments, size and access dates. The "Access" sheet 670 enables or disables the various controls that surround the body of a document. These options are generally used when embedding one document into another to

cause the embedded document to blend in with the surrounding main document.

Figure 15 shows a text document containing an embedded drawing document 680 with its surrounding border displayed. Figure 16 shows the same document with the border of the embedded document disabled, seamlessly merging the two documents. Like floating pages, option sheets can be dragged using the title bar 650 and re-sized by dragging the "handle" 630 on the borders (see Figure 12).

Option sheet stacks are used throughout out the present invention to control the attributes of many types of objects. For instance, the fonts and sizes of letters in a document's text can be controlled through a text option sheet 690, shown in Figure 17. Option sheets are available for such diverse objects as letters, paragraphs, pages, geometric figures, spreadsheet cells, graphical images, icons, or whole documents.

Referring to Figure 11, other commands on the "Document" menu will print the document (command 540); control the format of printing (command 550); send the document to a remote computer or device (command 570); export the document to a foreign file format (command 570); or revert to the previous version (command 580). The exact contents of the "Document" menu is dictated by the type of the document, but the "About" command 530 is always available.

The "Edit" menu 720, shown in Figure 18, contains commands which apply to objects selected from within the document. A selection is generally made by clicking or tapping on the target, or by dragging the cursor through many objects. For example, a text document is biased towards operations on words: double-clicking over a letter selects the entire word. Many words could be selected by dragging the cursor or pointing device through the desired words.

In text, the selection is indicated by a gray background 730. Other types of objects can use different indications, such as the small square handles 330 displayed by geometric figures, as in Figure 7.

The Edit menu 720 shown in Figure 18 typically provides commands to

move (command 740), copy (command 750) or delete (command 760) the selection, as well as to access (command 770) the option sheet stack that pertains to the selection. Other commands 780 specific to the objects found in different document types are located found on the "Edit" menu.

5 It is also possible to move or copy text within a document or to another document by using gestures. A target object is first selected by pressing the pen on the selection until an indication of the selection occurs, such as highlighting or an animated line surrounding an icon which shows the selected object (or a portion of it, if the selection is text longer than a few
10 words). In a "move", the selected wording is actually moved (which is to say, it is deleted from its original location); in a "copy", a copy of the selected wording is moved.

 The icon is then dragged to another location within the document or page being displayed, or to another location within the notebook to which the
15 user navigates. In the latter case, the user may turn to a different page, in which case the icon will follow the page turn, and then "drop" or release the icon (and hence the selected information) by lifting the pen at the new location. For instance, the user may turn to the table of contents and drag a target object such as a document to a location on the table of contents.

20 As indicated above, this procedure may be used either to drag a copy of the target, or to move the target, to the new location. The move and copy operations may be distinguished either on the basis of gestures, or by using the Edit menu. Thus, a move operation may be carried out by tapping the target three times in quick succession, followed by navigation to the new
25 location, dragging the target (or the icon thereof) to the new location, and dropping the copy there. Alternatively, the target may be selected, followed by a selection of "Move" from the Edit menu, and then dragging and dropping a copy of the target thereto. (If an entire document is moved, as noted earlier, its page number changes and the operating system of the invention
30 automatically resequences that documents and the other documents is the sequential order of documents.) A copy operation, on the other hand, may be a press-hold-drag-drop gesture, or a select-drag-drop gesture coupled with the

choice of "Copy" from the Edit menu after selection of the target.

Some specialized documents such as "Help" are better presented as notebooks separate from the main notebook. This separate-notebook format allows a page from the "Help" notebook 790 (see Figure 19) to be referenced as it floats above a user's document 800. Such independent notebooks support all of the features of the main notebook described above. A user might also organize different sets of documents into separate notebooks.

The "Contents" page of a notebook, shown in Figure 20, contains entries 810 for all top-level documents contained within that notebook. Each entry includes an icon 820, a name or title 830, a button 840, and a page number 850, all of which are automatically generated by the system of the invention, and which provide direct access to the various associated documents.

The contents page is itself the top-most instance of a document of type "Section". The entries 860 whose titles are bolded (Samples, Staff, etc.) are also "Section" documents which contain sub-sections of the notebook, and whose first pages constitute subsections of the Table of Contents. Each section includes a tab (such as the tab for "Widget Inc." shown in Figure 20), which both identifies the section and includes a subset of the tabs of the notebook. The tabs are automatically assigned the names of the pages which they identify, but can be renamed. If a document title or name is changed, and the associated tab has not been edited, the tab is likewise changed. However, if the tab label is edited and the document or page name is subsequently changed, this new change is not automatically implemented in the tab name.

The entries 870 whose titles are in a font of normal weight represent documents of other types. A page is selected by clicking on its title, and its selection is then indicated by a gray background 880 on the title.

Referring to Figure 21, an option for an expanded view (command 890) or collapsed view (command 900) can be selected from the "View" menu 930. In an expanded view (fig. 22) the nested contents 940 of the section will be displayed. Double-clicking on a section name will also expand or collapse that section, depending on its state.

5 If "Expand" or "Collapse" are chosen when no page is selected then all sections are affected. Figure 23 shows a portion of the table of contents with all sub-sections expanded. Note that a scroll margin 950 has been automatically added because the expanded list cannot be completely displayed in the window.

10 The pages of a notebook may be accessed in many ways. Adjacent pages may be turned to by clicking on the "next page" 440 or "previous page" 430 buttons, shown in Figure 10. Clicking on a document's page number 850 (see Figure 20) in the table of contents or icon 820 will turn to that page. Pages may also be accessed by selecting them and choosing "Bring-to" 920 or "Turn-to" 910 (see Figure 21) from the "View" menu 930.

15 Page tabs 960, shown in Figure 24, are visible for pages where a page tab has been enabled and can be clicked upon to turn to that page. The table of contents always has its page tab as the top-most tab 970. A page's optional tab is controlled by a check-box 980 which is made visible by selecting the "Tab Box" option 990 from under the "Show" menu 1000. When there is not enough room to display all the page tabs the lower tabs 1010 will be stacked. In order to see these hidden tabs a special view 1020 (see Figure 25) is available which shows all tabs. This view is obtained by flicking to the left on any of the hidden tabs. Any subsequent action will again collapse the tabs, and tapping on any given tab will turn to the referenced document.

20 The "Show" menu 1000 (shown in Figure 24) in a section allows individual control over what information is shown for all listed documents. Each item may be separately toggled by tapping on it. "Icon" 982 controls the display of the small icon defined for each document type. "Type" 984 when enabled will display the name of a documents type. "Date" 986 controls the display of the document's last modification date. "Size" 988 controls the display of the documents' filed (i.e., inactive) size. "Tab Box" 990 will display the page tab check boxes when enabled.

30 Where a single click causes a page to be turned to, a double click will cause that page 1030 to be floated above the current page. The turn-to/bring-to meaning for clicking and double-clicking is used in many places

throughout the notebook. The icons 820 and page numbers 850 in a section document, the tabs 960, the page numbers 1290 in the index document (see Figure 35) and "go to" link buttons 230 all support this behavior.

5 A floating page can be sent back to its home location by clicking on the triangular button 1040 (see Figure 26) located at its upper left corner. Floating pages can be expanded to the full size of the display, as shown in Figure 27, by an upward stroke in the title bar 1050, and can be reduced to their original floating size by a downward stroke on the title bar 1080 of the full size page.

10 In addition, floating pages can be re-sized through the "handles" 1060 and 1070 at the lower right and bottom of their frames, shown in Figure 26. The boundary 1090 of the floating page can then be shrunk or stretched by dragging the box to the desired position. Figure 28 shows the expanding boundary 1090, and Figure 29 shows the result. In a similar fashion the location of floating pages can be changed by dragging an outline 1100 (see 15 Figure 30) by means of the title bar to the new location 1110, shown in Figure 31. Floating objects maintain their positions as the underlying page is turned.

The "Edit" 1120 menu of a section document, shown in Figure 32, is used to move (command 1130), copy (command 1140), delete (command 1150) and rename (command 1160) individual pages.

20 New pages are added to the section by choosing a document type 1190, shown in Figure 33, from under the "Create" menu 1200. The new page is placed after, and in the same section as, the current selection. As shown in Figure 34, the pseudo-entry "--Empty--" 1180 may be selected to place a page in an empty section 1170. If there is no selection, then the new page is placed at 25 the end of the list of pages.

Although section documents normally display their contents in page-number order (as in the Table of Contents), the "Sort" menu 1210, shown in Figure 34, can provide other orders. Entries within a section can be listed by page (command 1220), name (i.e. alphabetically: command 1230), type 30 (command 1240), date (i.e. chronologically: command 1250) or size (command 1260). Thus, commands in the Sort menu let the user rearrange the Table of Contents by criteria similar to the Show menu. The default display order is

the order in which the documents appear in the notebook.

Sorting documents using the Sort menu rearranges their order in the Table of Contents, but in a preferred embodiment does not affect their actual sequential ordering as governed by the operating system. However, the
5 creation, deletion, insertion and moving of a document (i.e. transfer from one location to another) do cause the operating system to automatically reorder the pages to reflect such actions.

As in many books, an index 1270 (shown in Figure 35) may be found at the back of a notebook. Like its real-world counterpart, the index is an
10 alphabetical list of keywords or phrases 1280, and of the numbers 1290 of the pages on which they occur. The system of the invention automatically creates the index, and automatically modifies it as documents are created, moved, modified and deleted, to reflect the new sequential ordering of documents as managed by the operating system.

15 Keywords that appear on multiple pages list all occurrences 1290. The initial set of entries includes the titles of all the documents in the notebook. The user may add keys 1300 and 1310 from any words 382 and 384 (Figure 8) in any document in the notebook. Pages can be accessed by clicking or double-clicking on the page number to turn-to or float the page, respectively.

20 Link buttons allow the user to place ties from anywhere in the notebook to any other object in another document. Figure 36 shows a very useful application of link buttons. The main document is a long, frequently-referenced report. In this case, link buttons 1320 have been placed on the document's cork margin 1330, and these buttons have been tied to some of
25 the major headings 1340 in the report. Clicking on any of the buttons 1320 causes the document to scroll the target to the window.

Such a link button could also be placed: in the body of the document, as shown at numeral 1350 in Figure 37; or on the bookshelf 230; or in another document (see numeral 1360 in Figure 38). No matter where the button is
30 placed, clicking on it will cause the target to be turned to, while double clicking will float the target 1370 as shown in Figure 39.

Thus, generally speaking one document may be assigned to a first page,

and another document may be assigned to a second page, and the link button is created and assigned to the first page, i.e. the page with the button, which is displayed in the document display region. The link button is, in the example above, utilized while the first document is displayed in the document display region, unless it resides on the bookshelf, in which case (like all other items on the bookshelf) it is always available regardless of the document which is displayed or the application which is in use.

It is quite easy to use link buttons to create a "hypertext" document. Figure 40 shows a link button 1380 on the bookshelf tied to a word 1390 on a frequently-referenced page in the Help notebook 1400. When the button is pressed, the Help notebook will open and turn to the linked page.

From the above examples, it is seen that link buttons may either cause the target document (such as document 1370 in Figure 39) to be displayed in a second document display region, i.e. one that does not precisely coincide with the first document display region, in which the linking document is displayed. Alternatively, utilizing the link button may cause the target document to be turned to, i.e. displayed in the first document display region, replacing the contents of the first document.

20 Embedding And The Implementation of the Notebook Metaphor

Embedding is a fundamental concept in the implementation of the present invention. For the purposes of this application, embedding can be defined as the construction of compound documents by the nesting of autonomous documents within another document. This nesting can be carried out to any level. A simple example of this is in Figure 42, showing the construction of a document containing some text 1470 and 1480 and a diagram 1490. This document is constructed by embedding a drawing document into a text document. Without support for the embedding of independent disparate documents, the underlying text editor program would need to support the functions of a drawing program. With embedding, however, a user can construct compound documents from any available document, where each document may include an independently ordered sequence of pages.

Moreover, as shown in Figure 42, the embedding document may be created by one application, while the embedded document is created by a second, different application.

The notebook takes this structure to its logical end: the entire system is
5 built as one all-encompassing document. The root document is the bookshelf. All objects on the bookshelf are embedded in the bookshelf, including the main notebook. The sole document directly embedded in the main notebook is the table of contents document. The documents embedded in the table of contents are chosen by the user. They may be of any type of
10 documents, including multiple documents, compound documents or section documents which contain other documents. The embedded document is preferably displayed in its position as embedded in the embedding document in the document display region, or an icon of the embedded document may be displayed there instead. The text surrounding the embedded document or
15 its icon preferably flows or wraps around it, filling in the surrounding area in the document display region.

The notebook metaphor hides the existence of the computer programs which implement its behavior and that of the different types of documents. Other, earlier systems use a model in which a program is executed and can
20 load, modify, and save data files of an acceptable format. In the present invention, by way of contrast, all the user needs to see are documents. The execution of the correct program and the loading of a specific data file normally occurs when a document is turned-to, floated or opened from an icon. Thus, an embedded document may be displayed and modified by
25 floating it in a display region above the display region in which the main (embedding) document is displayed; or it may be modified in-place, i.e. where it resides in the embedding document. If it is floated, then it may partially cover up the display of the embedding document, but does not replace it in the first document display region.

30 As a general rule, a document may be floated over an underlying document appearing in the document display region in at least two manners: first, such that the floated document remains in its own display region when

the user turns away from the underlying document (such as to yet another, i.e. third, document); and second, such that, when the user turns away from the underlying document (i.e. to a third document), the floatedd document ceases to be displayed.

5 An identifier that uniquely identifies the type of the document is saved with the document's filed data when it is created or suspended. This identifier is used to select the appropriate program code to execute when the document is made active again.

10 The present invention implements embedding as part of the main operating system software, making embedding available to all application programs that execute within this "application framework". An application program need contain no code to deal with accepting embedded documents or being embedded itself. The application framework provides default behavior for embedded documents, but programs can override this default if desired.

15 The application framework is implemented on top of a preemptive, multi-tasking operating system which supports shared code, a tree-structured file system and a windowed graphics system. The support for embedding is built from the common "Operating System"-level objects. A non-compound (or simple) document is a separate task executing the shared code for that type of document. When the user turns away from a document's page or closes its window, the task is suspended and its state is saved in a sub-directory whose position in the directory tree corresponds to the position of the document's nesting within the bookshelf. This filed copy of the task's state includes more than just the data that makes up the document. Enough information is saved
20 such that when the task is re-started, the displayed page or window will appear exactly as it did when it last was displayed. To do this, the state of the window is also filed with the state of the document.

25 The application program displays its output and accepts pointing-device input through a window whose location and size are controlled by its parent application or (by default) the application framework. Thus, the application framework isolates an application program within its own task, directory and window. This isolation builds a very robust system without
30

burdening the user with an arcane command language. The user can locate, execute, combine, suspend, and destroy multiple tasks by simply manipulating pages in a notebook.

The bookshelf and its nested documents are constructed of standard components commonly found in tree-structured file systems, such as files and directories. The operations performed on these components (move, copy, delete) are also available in all functional file systems. Because of this, it is easy to map the structure of the present invention onto a foreign file system. If this external file system can be shared by multiple machines, then a number of machines implementing the present invention can share and exchange entire notebooks through the shared medium.

Consider the document shown in Figure 42. This compound document was created by embedding a drawing document 1490 within a text document called "Org Chart". Figure 41 shows how these documents are stored in the file system of the present invention. Each enclosing document is represented by a directory 1420, which contains its sub-documents 1410. At each level a file called "docstate.res" holds the state of the document when it is inactive. The text document "Org Chart" 1420 is embedded in a section document 1430 called "Widget Inc." which is, in turn, embedded in the table of contents document 1440. The table of contents is the sole top-level embedded document within the main notebook 1450. At the highest level the main notebook is embedded in the book shelf 1460.

Because each embedded document has its own task which executes when the document is visible, the drawing 1490 is not merely a static image, but is a window or ("edit region") into a drawing document controlled by an actively executing program. Thus if "Tom Keating" 1510 decides to be known by his middle name (see Figure 42), his "box" 1500 could be re-sized and his name edited in the edit box 1520 directly in the displayed compound document (where, in this case, the embedded document is created by a different application than the embedding document), to produce the revised version shown in Figure 44. To do this, it is not necessary to select the embedded document or switch to another program; that is, when the

embedding document is opened, the embedded document (here, the organization chart) is automatically opened in an edit region at which it can be edited, using its own application (i.e. the drawing application, as opposed to the text application of the embedding document).

5 In the above example, the modification of the embedded document takes place by means of alterations made with the embedded document in place within the embedding document. Likewise, other actions may be performed on the embedding document which are also automatically executed on the embedded document, such as printing, copying, deleting,
10 closing (i.e. ceasing display) and saving, and so on. In the case of printing, an embedded document may be printed by itself by selecting "Print" from the embedded document's "Document" menu, or the entire embedding document may be printed along with the embedded document by selecting "Print All Contents" from the embedding document's Print Setup option
15 sheet. Yet another alternative is to print the embedding document only with so much of the embedded document as is displayed on screen. In each case, the print action is executed with respect to the embedded document automatically when the embedding document is printed.

The application framework allows applications to tell the embedded
20 objects when to file their data. This is done automatically, without the user's specific command; thus, when a user turns away from a page, the application framework tells the application to save its instance data, which includes any data in an embedded document. Similarly, when the larger (i.e. embedding) document is printed, copied, deleted or turned away from, the same action is
25 automatically taken on the embedded document. In the case of printing a document, the displayed portion of the embedded document is also printed. Alternatively, the entire embedded document may be automatically printed along with the embedding document, by selecting Print All Contents from the Print Setup options sheet.

30 Although the application framework provides default behavior for embedded documents, parent applications can override these defaults. For example, the bookshelf supports its embedded documents in two states: first,

as icons 120 (see Figure 1) arranged from left to right, bottom to top; and second, as floating windows, as the "help" notebook 790 shows in Figure 19.

A notebook causes its "children" to appear as a stacked series of numbered pages with a current page and any number of floating pages visible.

5 A section document presents its children as an indented list 940, as in Figure 22. A text document places its embedded document 1530 in-line with the text, as in Figure 45. Opening or re-sizing the in-line document 1530 causes the surrounding text 1540 to "flow" (or "wrap around") to fill in to best fit the embedded document.

10 To produce a "seamless" document the "Access" option card 1558 shown in Figure 47, is opened by using the "About..." command from the "Document" menu 1580. Setting the "Borders" control 1560 to "Off" and clicking on "Apply and Close" 1570 will remove the frame 1550 that surrounds the embedded drawing. This produces the final document shown
15 in Figure 48. Thus, there are two border options: the first option displays the border around the contents of the embedded document, and the second option displays the contents of the embedded document with no visible border.

Some embedded objects do not need the full capabilities of a document.
20 Such "components" allow for simpler objects with less overhead. Components do exist in their own window but do not execute in a separate task and file their data in the directory of their containing document. A link button is an example of a component. It contains three pieces of information which, taken together, uniquely identifies the target of the link. First is the
25 unique ID of the parent document that contains the target. Second is the ID of the component or sub-document that contains the target. Third is a unique token assigned by the immediate container to the target object when the link button was created. For example, within text this unique ID is simply the offset of the target character from the start of the text. When a link button is
30 selected, the operating system uses the document ID to cause the targeted document to be visible, and passes the other two IDs to the document's underlying application program. The application program in turn ensures

that the target is visible within the document's current window.

5 The normal behavior for an application is to be suspended when its window is closed or not visible. This default can be overridden by the application when necessary. For example the "out box" is a form of device output queue manager. The out box task executes without displaying a window, moving queued data to devices such as printers and modems. An application can also start its own hidden sub-tasks. For example a spreadsheet could run a task to recalculate in the background while the main task handled user interface events.

10 The nested structure of documents and background execution can also be exploited by specialized "traversal" tasks which need to "walk" through branches of the tree to process all the simple documents in a compound document. Examples of operations performed by such tasks are "search and replace", "bulk spell checking", and printing. These invisible tasks will start a
15 sub-task for each sub-directory in-turn and issue specialized requests to the executing application code.

What is claimed is:

1. A computer having a tree-structured file system supporting nested
5 sub-directories, including means for recording relationships of compound documents by representing each document as a sub-directory.

2. A file system for a computer, having document page numbers
10 which are automatically generated by a user's placement of each document relative to other documents stored on the computer.

3. A computer system having a bit-mapped, graphical display and a pointing device, and including means for manipulating, modifying and combining computer documents, and further including:
15 a multipage notebook, wherein each said page is for controlling a computer document or file, and a unique page number is associated with each said page;

means for filing document data and document state whenever a user turns away from a page or closes a window; and
20 means for restoring said document data and document state when the user turns back to the page or opens the window.

4. An apparatus for controlling a computer system, including:
means coupled to said computer for creating a plurality of pages;
25 means coupled to said computer for assigning each of a plurality of documents created by at least one application to one of said pages; and
means coupled to said computer for sequentially ordering said plurality of pages.

5. The apparatus of claim 4, wherein said plurality of documents includes a first document created by a first application and a second document created by a second application.
30

6. The apparatus of claim 4, wherein said sequential ordering means
comprise means for sequentially ordering a first and second plurality of said
pages, wherein said ordering of said first plurality of pages is independent of
5 said ordering of said second plurality of pages.

7. The apparatus of claim 6, further comprising means for embedding
into a page of said first sequentially ordered plurality of pages said second
sequentially ordered plurality of pages.
10

8. The apparatus of claim 4, wherein said sequential ordering means
includes automatic reordering means for reordering said plurality of pages
upon a modification to said sequentially ordered plurality of pages.

9. The apparatus of claim 8, wherein said modification to the order of
said sequentially ordered plurality of pages comprises the insertion of a page.
15

10. The apparatus of claim 8, wherein said modification to the order of
said sequentially ordered plurality of pages comprises the deletion of a page.
20

11. The apparatus of claim 8, wherein said modification to the order of
said sequentially ordered plurality of pages comprises the transfer of a page
from one location in said sequential order to another.

12. The apparatus of claim 4, further comprising a page identifier
associated with each of said plurality of pages.
25

13. The apparatus of claim 12, wherein said page identifier is a page
number indicating the relative order of the page with which it is associated in
said sequentially ordered plurality of pages.
30

14. The apparatus of claim 12, wherein said page identifier is a name.

15. The apparatus of claim 14, wherein said page identifier name is the name of the main document on said associated page in which all other documents on said page are embedded.

5

16. The apparatus of claim 4, further comprising an area within said sequentially ordered plurality of pages in which objects can be embedded and made available to the user of said computer from any of said plurality of pages.

10

17. The apparatus of claim 4, further comprising an area within each of said sequentially ordered plurality of pages in which objects can be embedded and made available to the user of said computer from said page.

15

18. The apparatus of claim 4, further including means for automatically generating a compilation of page identifiers.

19. The apparatus of claim 18, wherein said compilation of page identifiers also comprises a compilation of document identifiers for providing direct access to said plurality of documents.

20

20. The apparatus of claim 18, wherein said compilation of page identifiers comprises page numbers associated with said pages.

25

21. The apparatus of claim 18, wherein said compilation of page identifiers comprises names associated with said pages.

22. The apparatus of claim 18, wherein said compilation of page identifiers comprises document types associated with documents assigned to said pages.

30

23. The apparatus of claim 18, wherein said compilation of page

identifiers comprises dates associated with said pages.

24. The apparatus of claim 18, further comprising means for sorting said page identifiers.

5

25. The apparatus of claim 24, wherein said sorting means comprises means for sorting said page identifiers numerically.

10 26. The apparatus of claim 24, wherein said sorting means comprises means for sorting said page identifiers alphabetically.

27. The apparatus of claim 24, wherein said sorting means comprises means for sorting said page identifiers chronologically.

15 28. The apparatus of claim 18, wherein said compilation of page identifiers comprises a table of contents.

20 29. The apparatus of claim 28, wherein said table of contents is the first of said sequentially ordered plurality of pages.

30. The apparatus of claim 28, wherein at least one of said page identifiers identifies a section, the first page of which comprises a subset of said table of contents.

25 31. The apparatus of claim 28, wherein said sequential ordering means includes automatic reordering means for reordering said compilation of page identifiers in said table of contents upon a modification to said sequentially ordered plurality of pages.

30 32. The apparatus of claim 18, wherein said compilation of page identifiers comprises an index.

33. The apparatus of claim 18, wherein said compilation of page identifiers comprises tabs associated with said pages, said tabs made available to the user from any page.

5 34. The apparatus of claim 33, wherein said tabs are ordered in accordance with said sequentially ordered plurality of pages.

35. The apparatus of claim 33, wherein said sequential ordering means includes automatic reordering means for reordering said tabs upon a
10 modification to the order of said sequentially ordered plurality of pages.

36. The apparatus of claim 33, wherein at least one of said tabs identifies a section, the first page of which comprises a subset of said compilation of page identifiers.
15

37. The apparatus of claim 33, wherein said tabs are automatically assigned names associated with the pages they identify, further comprising means for renaming said tabs.

20 38. The apparatus of claim 33, further comprising means for reordering said tabs.

39. The apparatus of claim 4, said computer system also comprising a screen for displaying information, further comprising means for displaying a
25 page of said sequentially ordered plurality of pages in a page display region on the screen.

40. The apparatus of claim 4, said computer system also comprising a screen for displaying information, further comprising:
30 a first document assigned to a page of said sequentially ordered plurality of pages;
 means coupled to said computer for embedding into said first

document a second document, said first and second documents created by at least one application; and

means for automatically displaying on said screen, upon opening said first document, contents of said second document in an edit region at which
5 said second document can be edited using the application which was used to create said second document.

41. The apparatus of claim 4, said computer system also comprising a screen for displaying information, further comprising:

10 a first document assigned to a first page in said sequentially ordered plurality of pages and a second document assigned to a second page in said sequentially ordered plurality of pages;

means coupled to said computer for displaying contents of said first document in a first document display region on said screen;

15 means coupled to said computer for creating a link between said first document and said second document;

means coupled to said computer for assigning said link to said first document; and

20 means coupled to said computer for performing an operation utilizing said link while the contents of said first document are displayed on the screen in said first document display region.

42. The apparatus of claim 41, wherein said performing means includes means for performing a link operation comprising replacing the
25 display, in said first document display region, of the contents of said first document with contents of said second document.

43. The apparatus of claim 41, wherein said performing means includes means for performing a link operation comprising displaying the
30 contents of said second document in a second document display region.

44. The apparatus of claim 41, wherein said performing means

includes means for performing a first link operation comprising replacing the display, in said first document display region, of the contents of said first document with contents of said second document, and for performing a second link operation comprising displaying the contents of said second document in a second document display region.

45. The apparatus of claim 4, said computer system also comprising a screen for displaying information, further comprising:
a first document assigned to a first page in said sequentially ordered plurality of pages;
means coupled to said computer for displaying contents of said first document in a document display region on said screen;
means coupled to said computer for selecting a target object comprising a subset of said displayed contents, said target object displayed at a target location on said screen in said document display region;
means for navigating to a destination in one of said plurality of documents, said navigating means comprising means for displaying in said document display region contents of the document containing said destination which comprise said destination and which were not displayed in said document display region immediately prior to said navigation; and
means coupled to said computer for performing one of a first and second transfer operations on a selected target object, said first transfer operation moving said selected target object to a destination of said navigation, said second transfer operation moving a copy of said selected target object to said destination, wherein each of said transfer operations comprises means for displaying said selected target object at said target location.

46. An apparatus for controlling a computer system, said computer system comprising a screen for displaying information, including:
means coupled to said computer for embedding into a first document a second document, said first and second documents created by at least one

application; and

means for automatically displaying on said screen, upon opening said first document, contents of said second document in an edit region at which said second document can be edited using the application which was used to
5 create said second document.

47. The apparatus of claim 46, wherein said first document is created by a first application and said second document is created by a second application.
10

48. The apparatus of claim 46, wherein said embedding means includes means for embedding multiple documents into said first document.

49. The apparatus of claim 46, further comprising means for displaying
15 in a document display region on said screen contents of said first document and contents of said second document as embedded in said first document.

50. The apparatus of claim 46, further comprising means for displaying
20 in a document display region on said screen contents of said first document and an icon representing said second document.

51. The apparatus of claim 49, wherein the displayed contents of said second document are in line with the contents of said first document, the displayed contents of said first document being wrapped around the contents
25 of said second document.

52. The apparatus of claim 51, further comprising means for selecting, for said second document, one of a first and second border display options, wherein the first border display option displays a border around the contents
30 of said second document, and wherein the second border display option displays the contents of said second document with no visible border.

53. The apparatus of claim 49, wherein the displayed contents of said second document are displayed on top of a subset of the displayed contents of said first document, and cover up the display of said subset of the displayed contents of said first document.

5

54. The apparatus of claim 53, further comprising means for selecting, for said second document, one of a first and second border display options, wherein the first border display option displays a border around the contents of said second document, and wherein the second border display option
10 displays the contents of said second document with no visible border.

55. An apparatus for controlling a computer system, said computer system comprising a screen for displaying information, including:
means coupled to said computer for embedding into a first document a
15 second document, said first and second documents created by at least one application;

means for displaying on said screen contents of said second document in an edit region at which said second document can be edited using the application which was used to create said second document; and
20 means for automatically executing an action on said second document in said edit region upon executing said action on said first document.

56. The apparatus of claim 55, wherein:
said displaying means includes means for displaying said first
25 document on said screen; and
said executing means includes means for automatically ceasing display of said second document in said edit region upon ceasing of display of said first document.

30 57. The apparatus of claim 55, further including means for printing said first document, and wherein said executing means includes means for automatically printing said second document upon printing of said first

document.

58. The apparatus of claim 55, further including means for copying said first document, and wherein said executing means includes means for
5 automatically copying said second document upon copying of said first document.

59. The apparatus of claim 55, further including means for deleting said first document, and wherein said executing means includes means for
10 automatically deleting said second document upon deleting of said first document.

60. The apparatus of claim 55, wherein said first document is created by a first application and said second document is created by a second
15 application.

61. The apparatus of claim 55, wherein said embedding means includes means for embedding multiple documents into said first document.

20 62. The apparatus of claim 55, further comprising means for displaying in a document display region on said screen contents of said first document and contents of said second document as embedded in said first document.

25 63. The apparatus of claim 55, further comprising means for displaying in a document display region on said screen contents of said first document and an icon representing said second document.

30 64. The apparatus of claim 62, wherein the displayed contents of said second document are in line with the contents of said first document, the displayed contents of said first document being wrapped around the contents of said second document.

65. The apparatus of claim 64, further comprising means for selecting, for said second document, one of a first and second border display options, wherein the first border display option displays a border around the contents of said second document, and wherein the second border display option
5 displays the contents of said second document with no visible border.

66. The apparatus of claim 62, wherein the displayed contents of said second document are displayed on top of a subset of the displayed contents of said first document, and cover up the display of said subset of the displayed
10 contents of said first document.

67. The apparatus of claim 53, further comprising means for selecting, for said second document, one of a first and second border display options, wherein the first border display option displays a border around the contents
15 of said second document, and wherein the second border display option displays the contents of said second document with no visible border.

68. An apparatus for controlling a computer system, said computer system comprising a screen for displaying information, including:
20 means coupled to said computer for displaying contents of a first document in a first document display region on said screen;
means coupled to said computer for creating a link between said first document and a second document;
means coupled to said computer for assigning said link to said first
25 document; and
means coupled to said computer for performing an operation utilizing said link while the contents of said first document are displayed on the screen in said first document display region, said performing means including
means for performing one of a first operation and a second operation,
30 wherein said first operation is for replacing the display, in said first document display region, of the contents of said first document with contents of said second document, and wherein said second operation is for displaying the

contents of said second document in a second document display region.

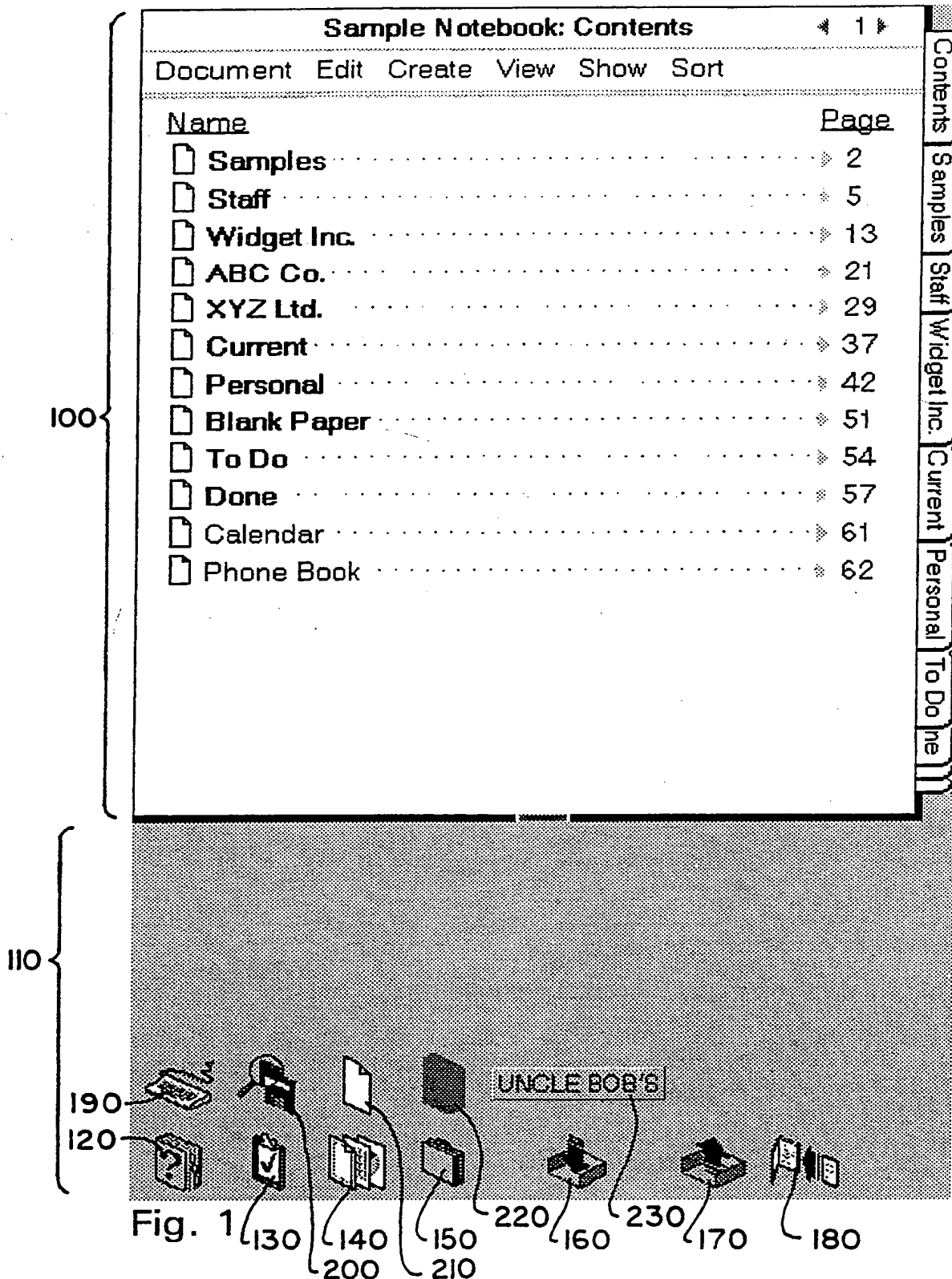
69. The apparatus of claim 68, wherein said displaying means includes:
means for replacing said first document in said first document display
5 region with a third document; and
means for maintaining the display of said second document upon
replacement of said first document by said third document.

70. The apparatus of claim 68, wherein said displaying means includes:
10 means for replacing said first document in said first document display
region with a third document; and
means for ceasing the display of said second document upon
replacement of said first document by said third document.

71. An apparatus for controlling a computer system, said computer
15 system comprising a screen for displaying information, including:
means coupled to said computer for displaying one of a first document
portion and a second document portion in a document display region on said
screen;
20 means coupled to said computer for selecting a target object comprising
a subset of said first document portion when said first document portion is
displayed;
means for navigating to a destination in said second document
portion, said navigating means comprising means for displaying said second
25 document portion in said document display region in an overlapping fashion
with said first document portion, thereby creating an overlapping display
region; and
means coupled to said computer for performing one of a first and
second transfer operations on a said target object, said first transfer operation
30 for moving said selected target object to said destination, and said second
transfer operation for moving a copy of said selected target object to said
destination;

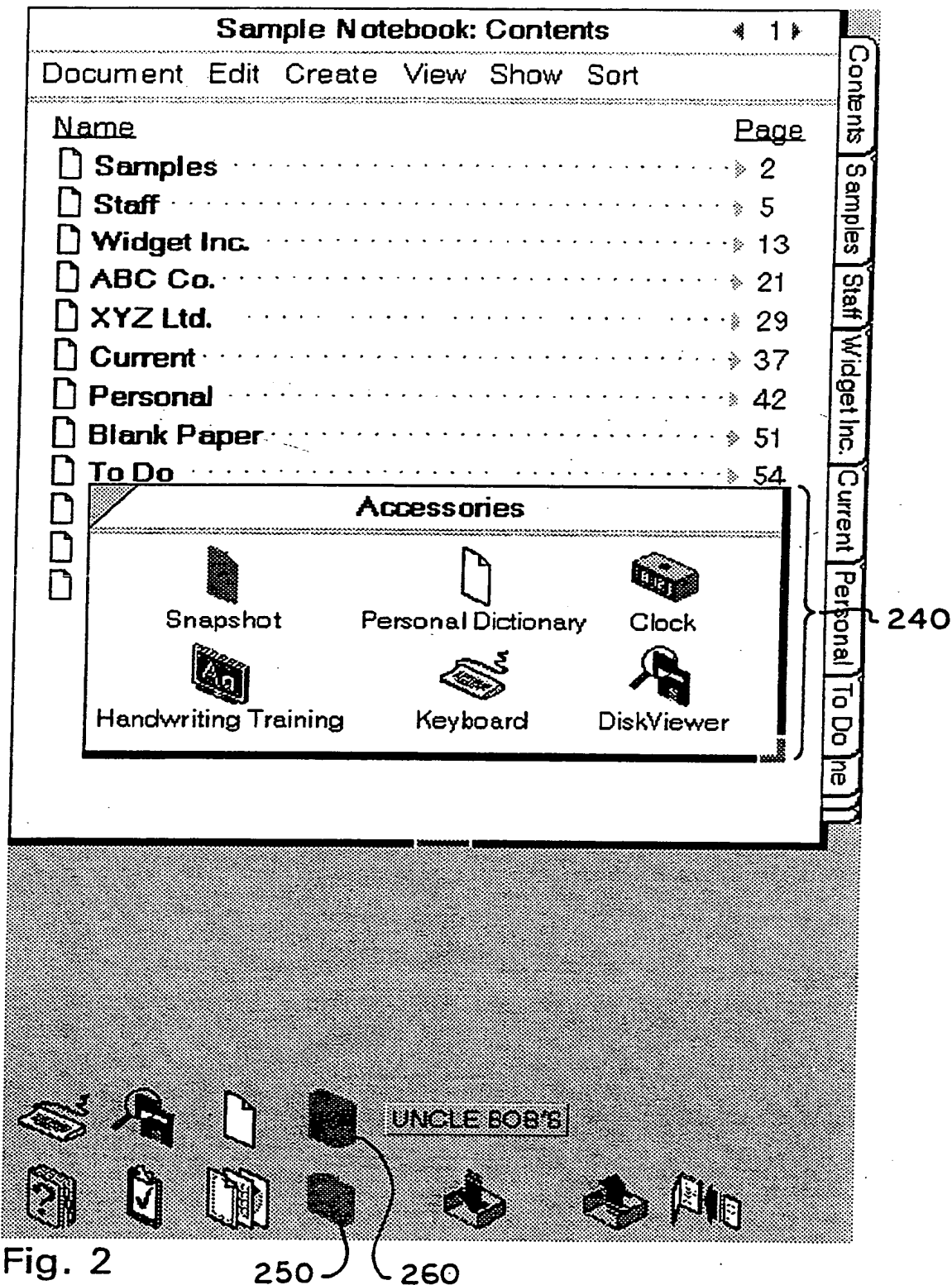
wherein said displaying means includes means for maintaining display of said target object when said target object is located within said overlapping region.

01/48



SUBSTITUTE SHEET

02/48



SUBSTITUTE SHEET

03/48

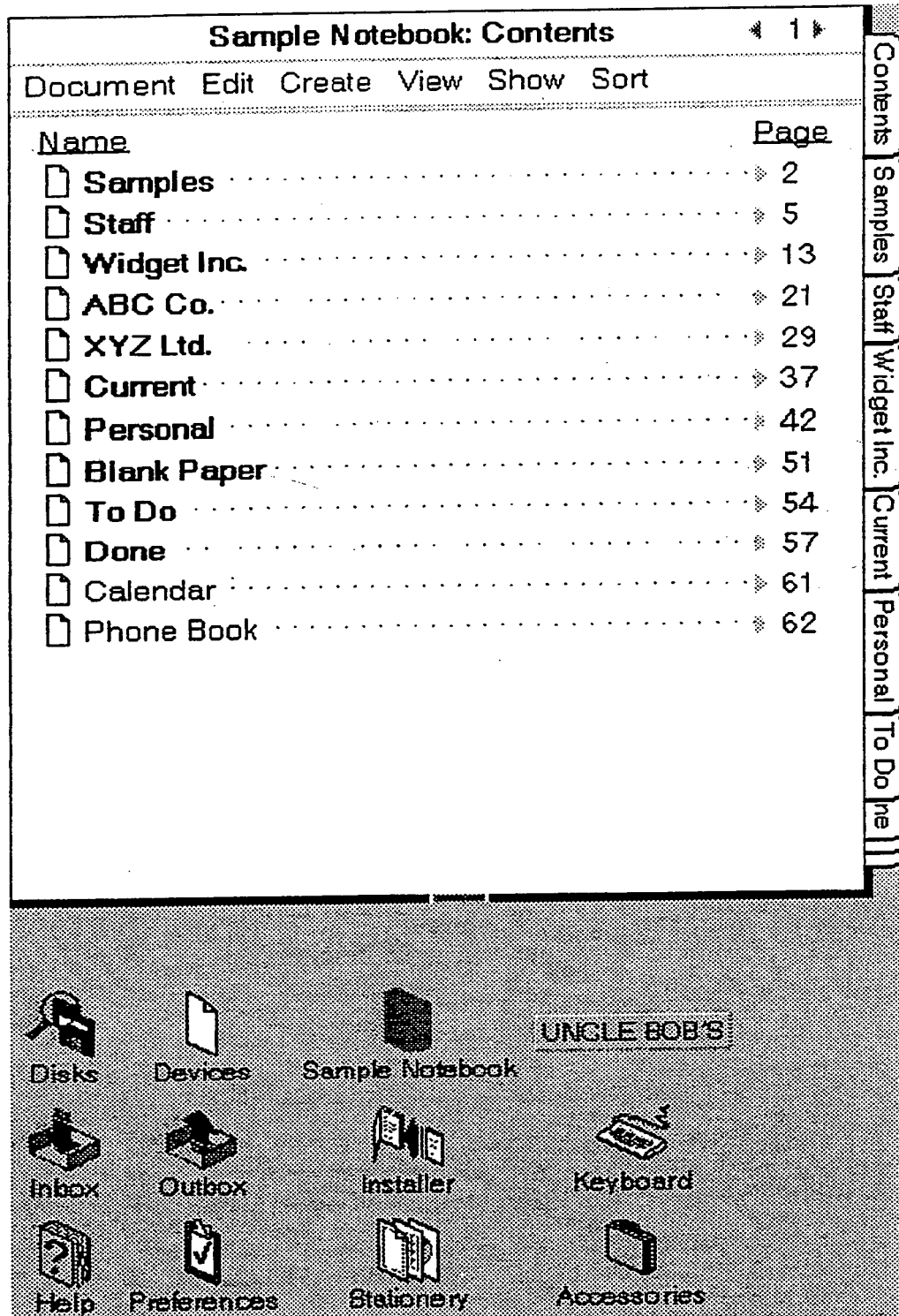


Fig. 3

SUBSTITUTE SHEET

04/48

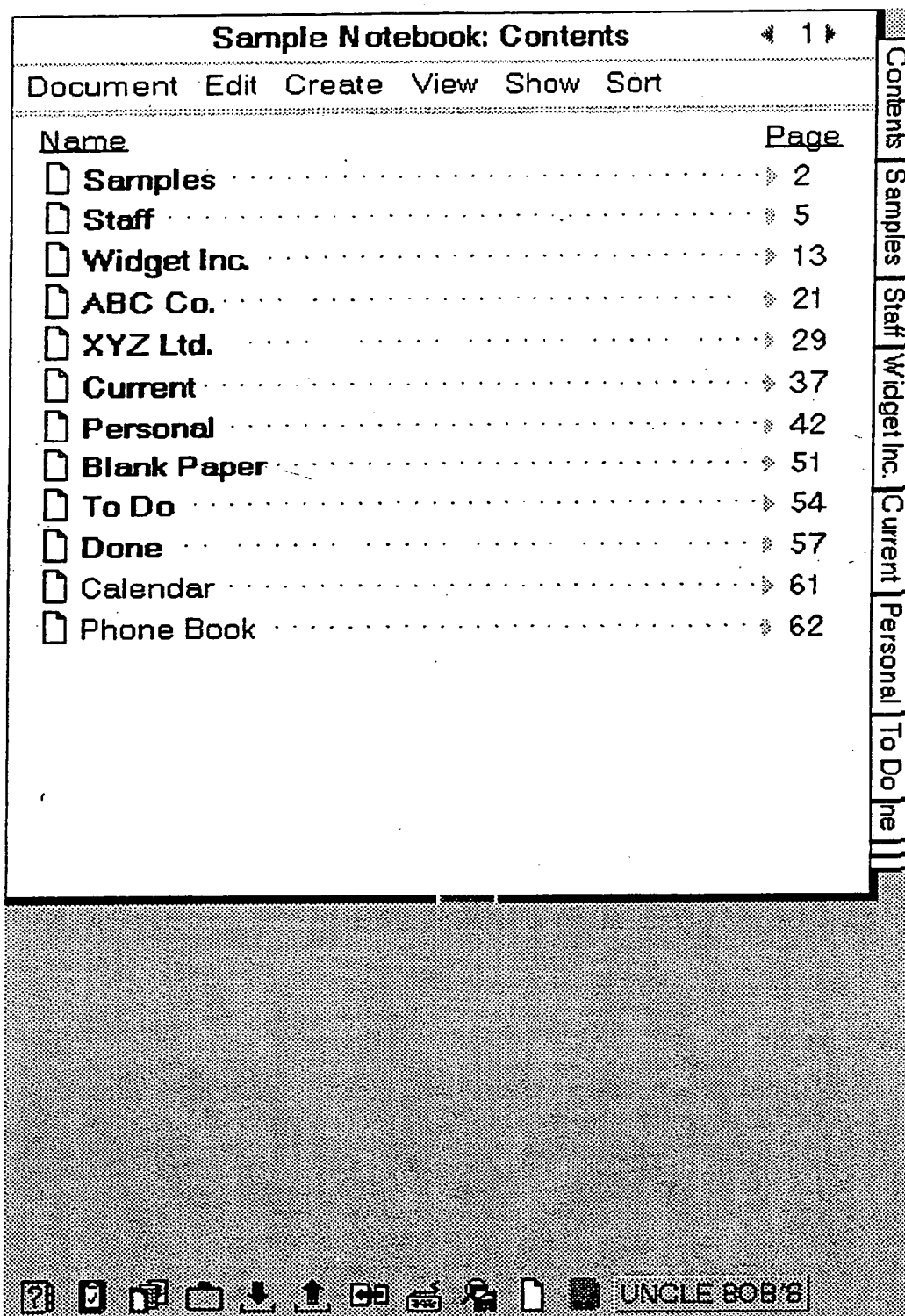


Fig. 4

SUBSTITUTE SHEET

05/48

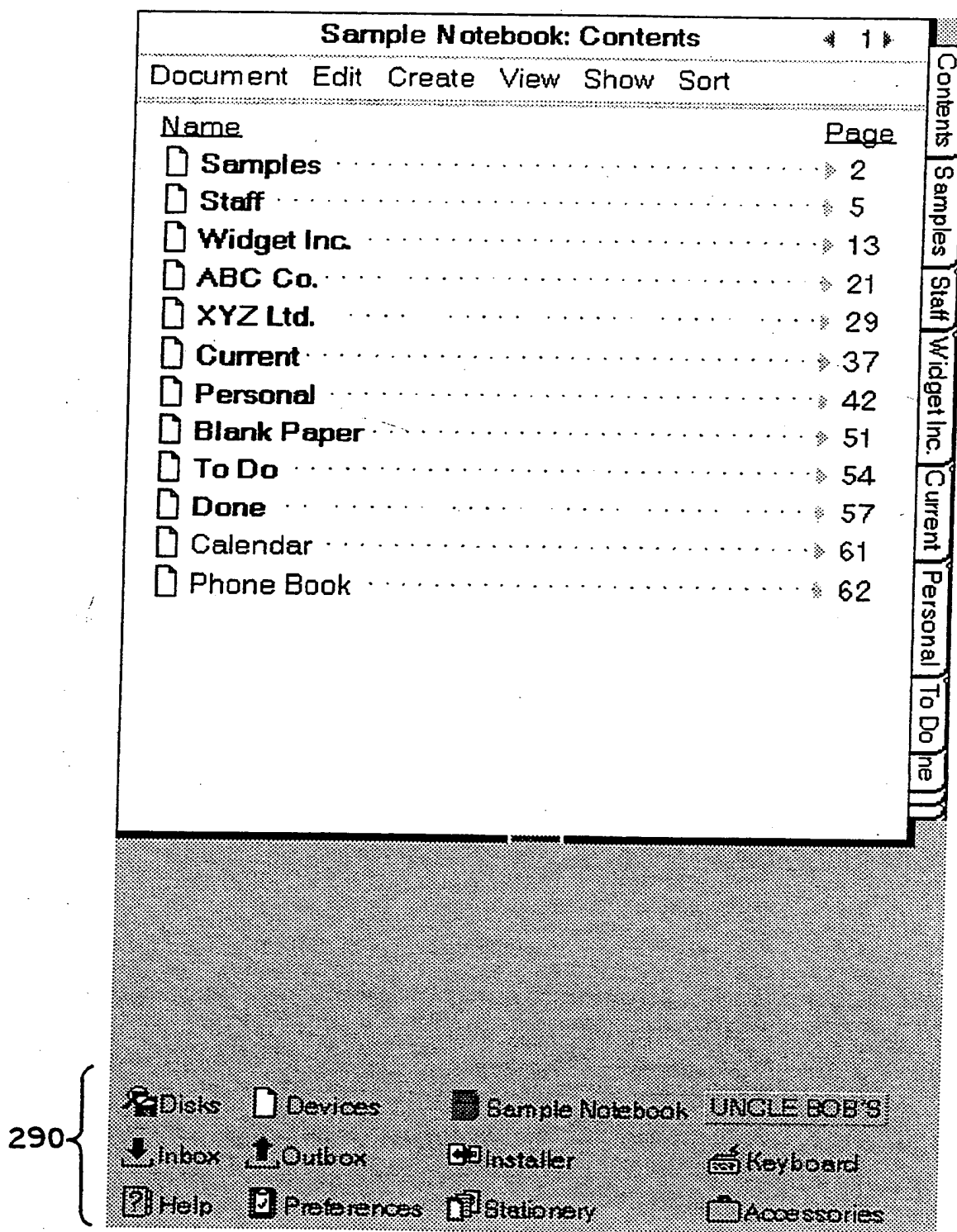


Fig. 5

SUBSTITUTE SHEET

06/48

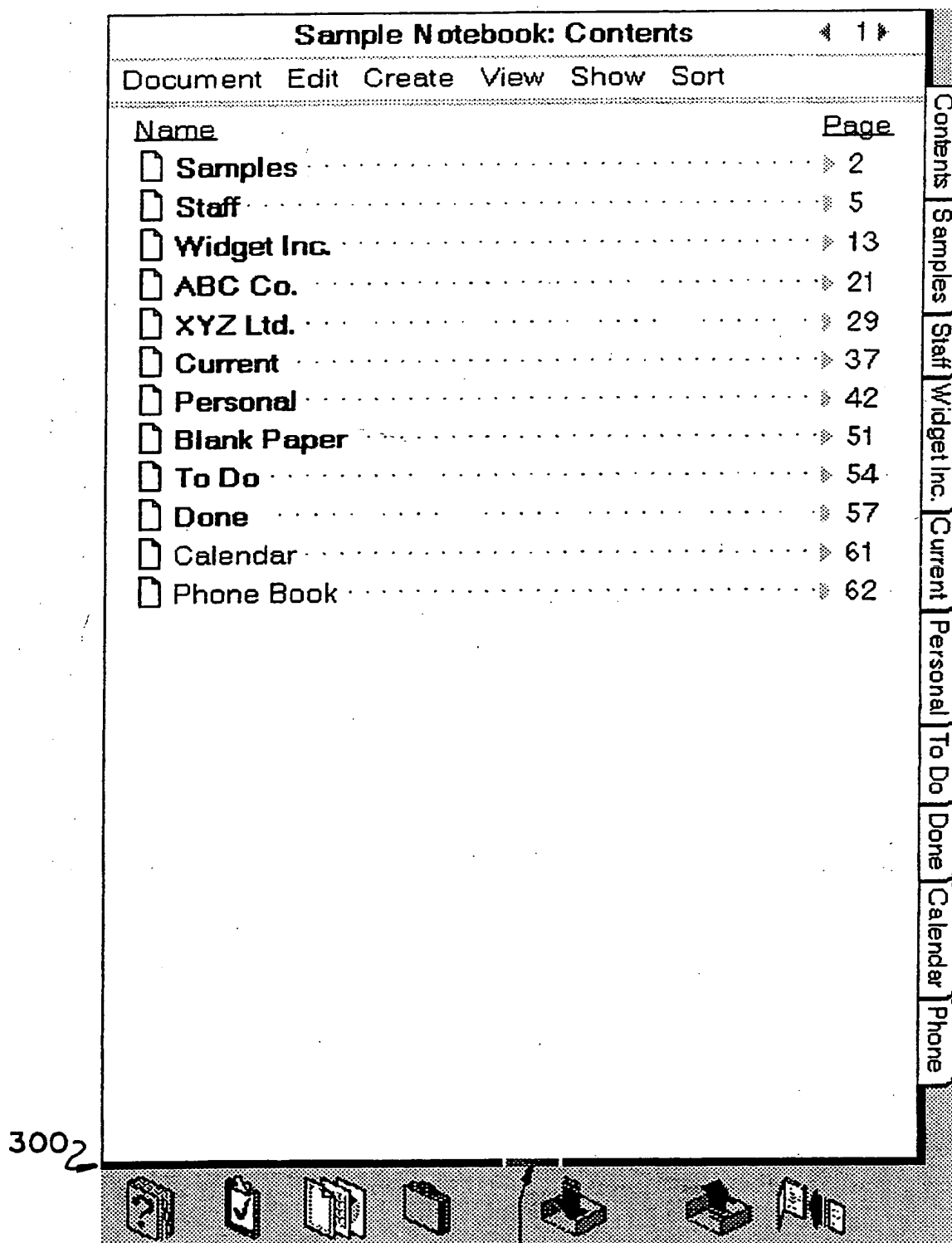
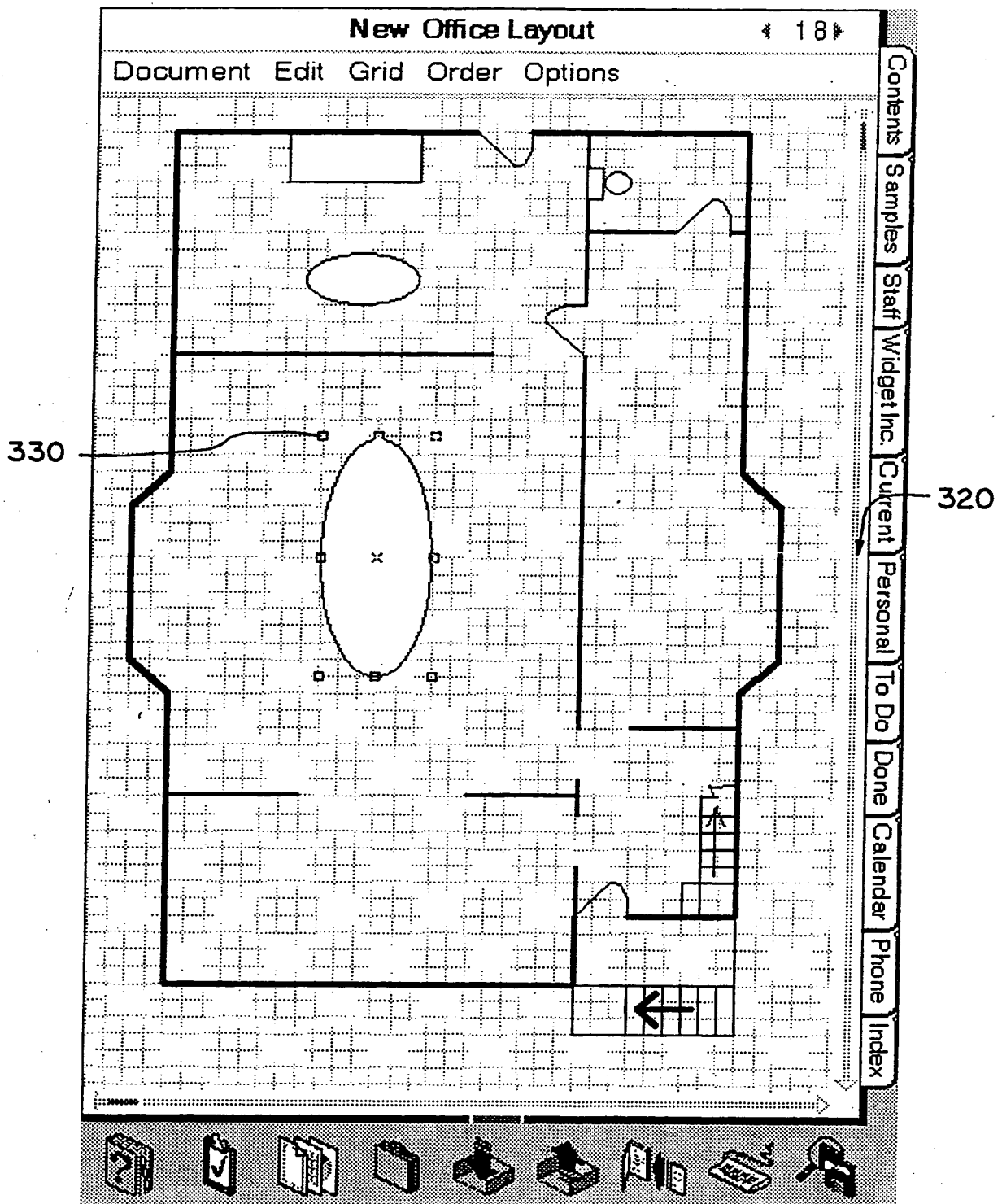


Fig. 6

SUBSTITUTE SHEET

07/48



SUBSTITUTE SHEET

08/48

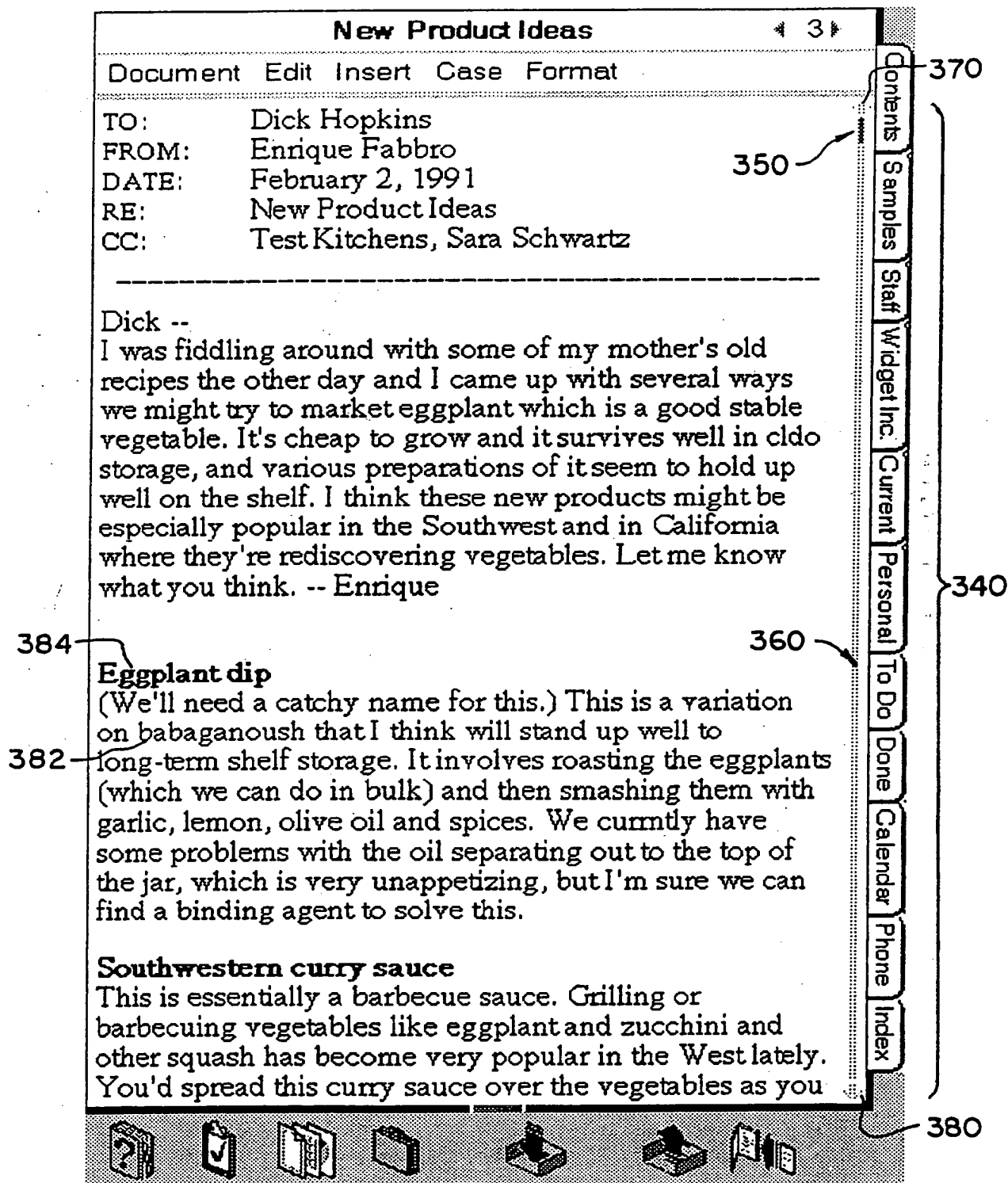


Fig. 8

SUBSTITUTE SHEET

09/48

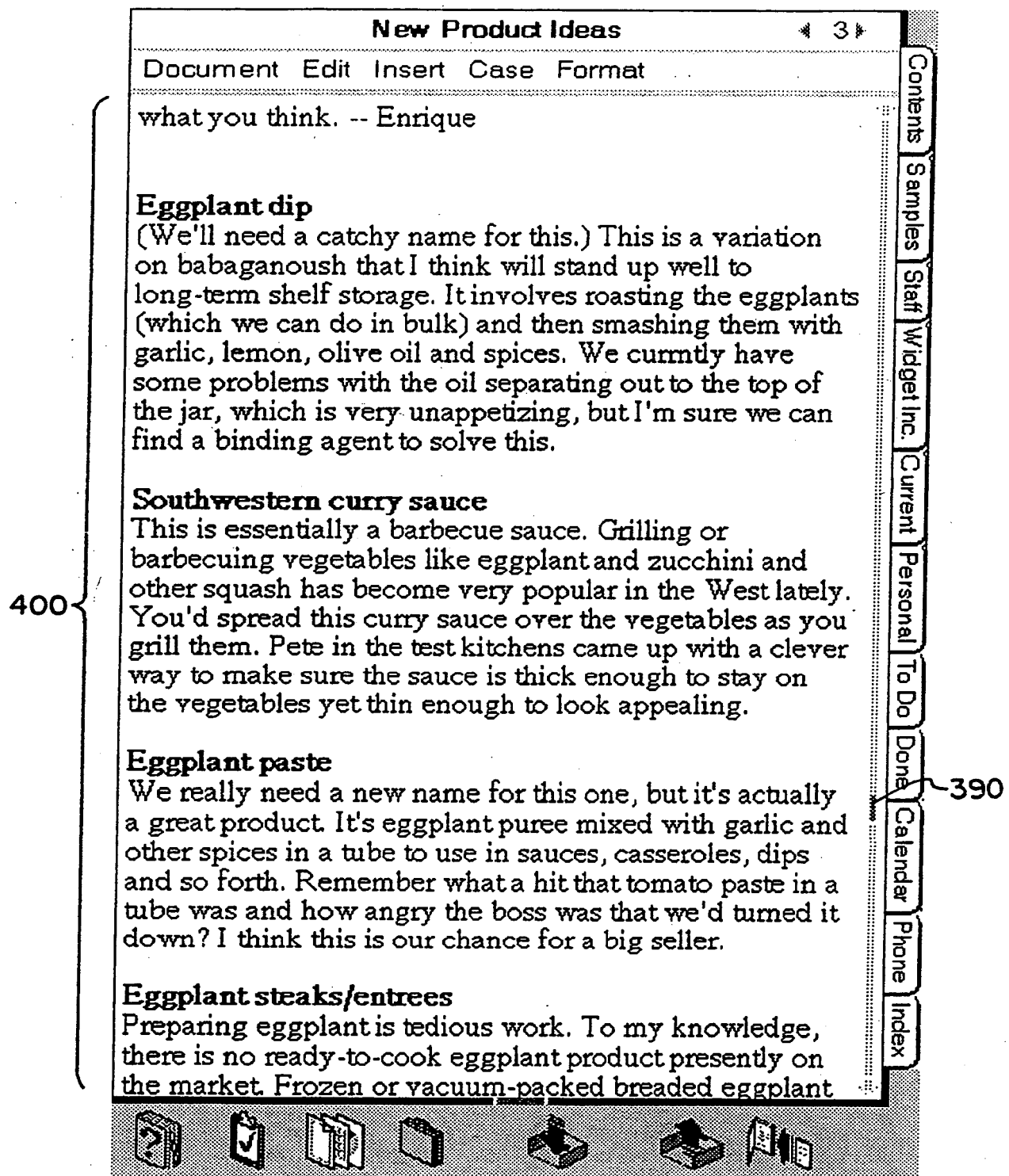
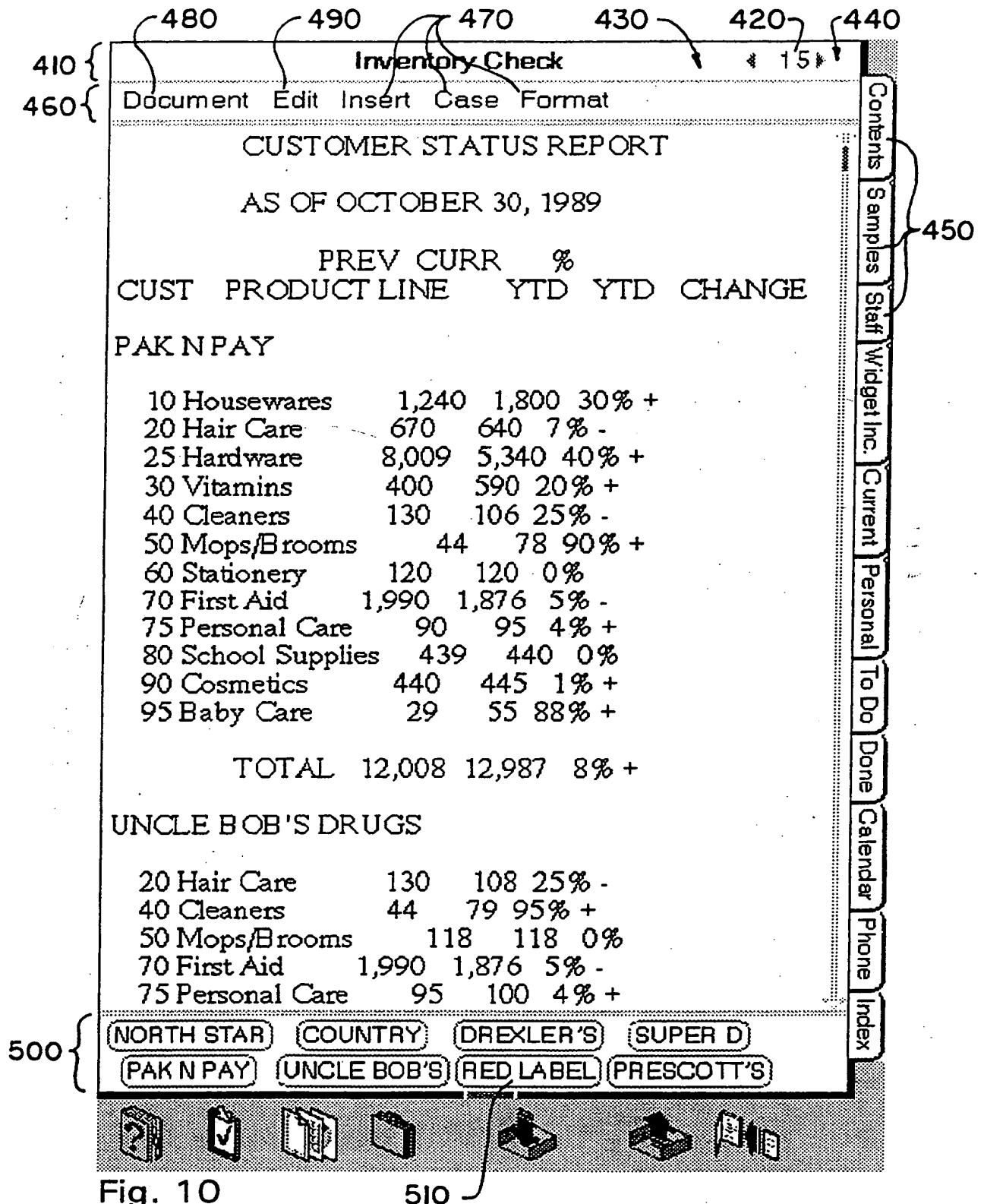


Fig. 9

SUBSTITUTE SHEET

10/48



SUBSTITUTE SHEET

11/48

Inventory Check 15

Document Edit Insert Case Format

About... 530
 Print... 540
 Print Setup... 550
 Send... 560
 Export... 570
 Revert 580

CUSTOMER STATUS REPORT

OF OCTOBER 30, 1989

	PREV	CURR	%
10 Housewares	1,240	1,800	30% +
20 Hair Care	670	640	7% -
25 Hardware	8,009	5,340	40% +
30 Vitamins	400	590	20% +
40 Cleaners	130	106	25% -
50 Mops/Brooms	44	78	90% +
60 Stationery	120	120	0%
70 First Aid	1,990	1,876	5% -
75 Personal Care	90	95	4% +
80 School Supplies	439	440	0%
90 Cosmetics	440	445	1% +
95 Baby Care	29	55	88% +
TOTAL	12,008	12,987	8% +

UNCLE BOB'S DRUGS

20 Hair Care	130	108	25% -
40 Cleaners	44	79	95% +
50 Mops/Brooms	118	118	0%
70 First Aid	1,990	1,876	5% -
75 Personal Care	95	100	4% +

NORTH STAR
COUNTRY
DREXLER'S
SUPER D

PAK N PAY
UNCLE BOB'S
RED LABEL
PRESCOTT'S

Fig. 11

SUBSTITUTE SHEET

12/48

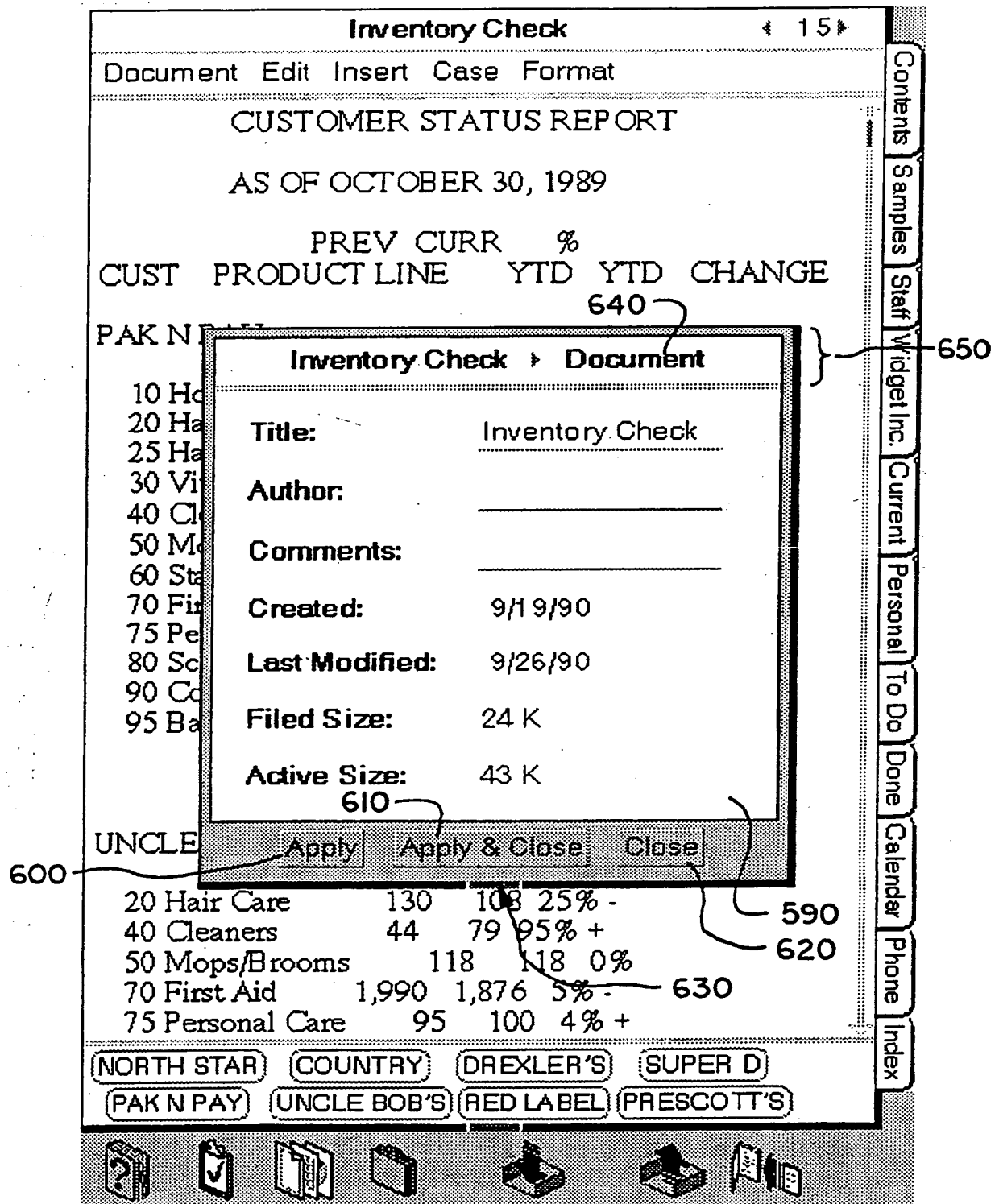


Fig. 12

SUBSTITUTE SHEET

13/48

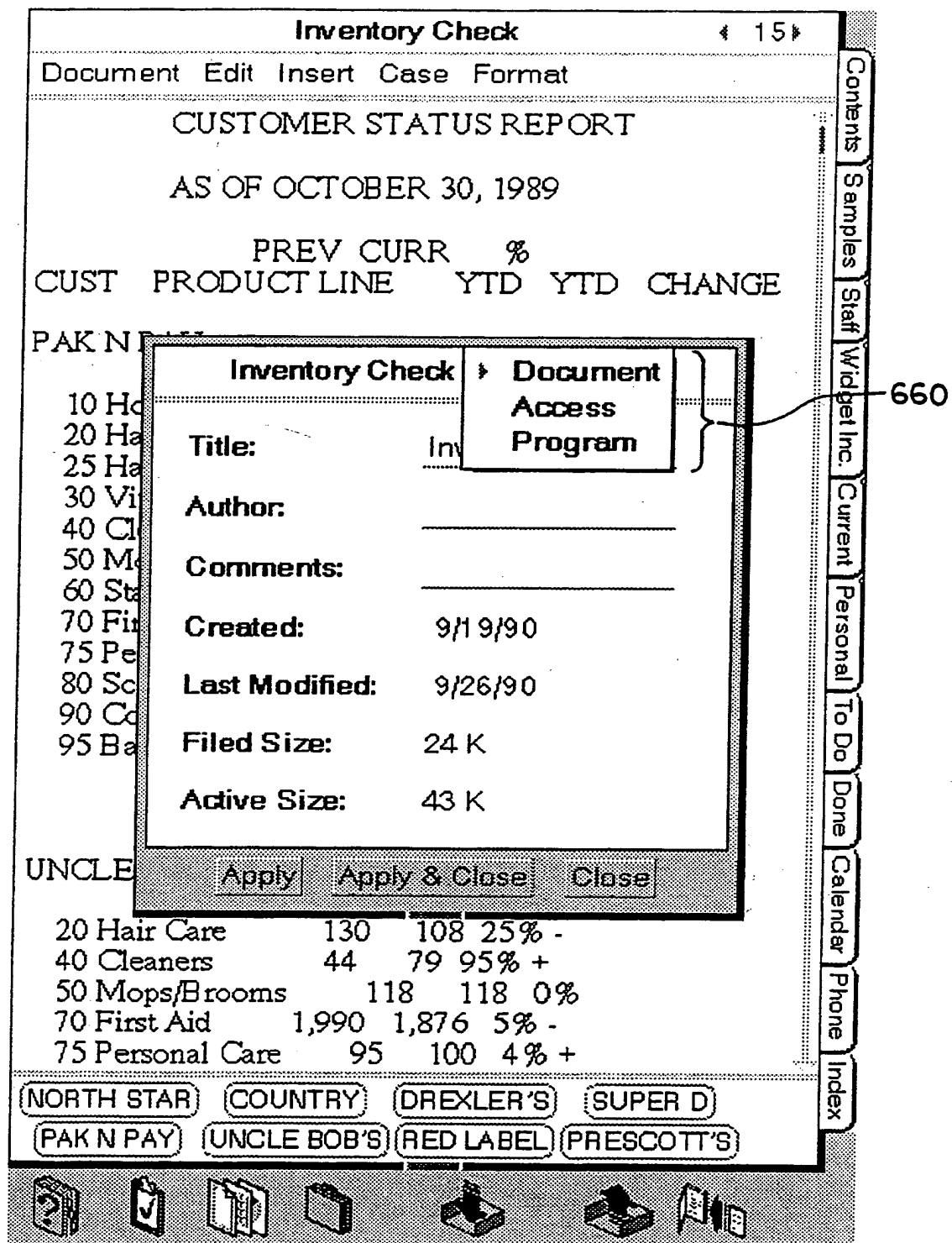


Fig. 13

SUBSTITUTE SHEET

14/48

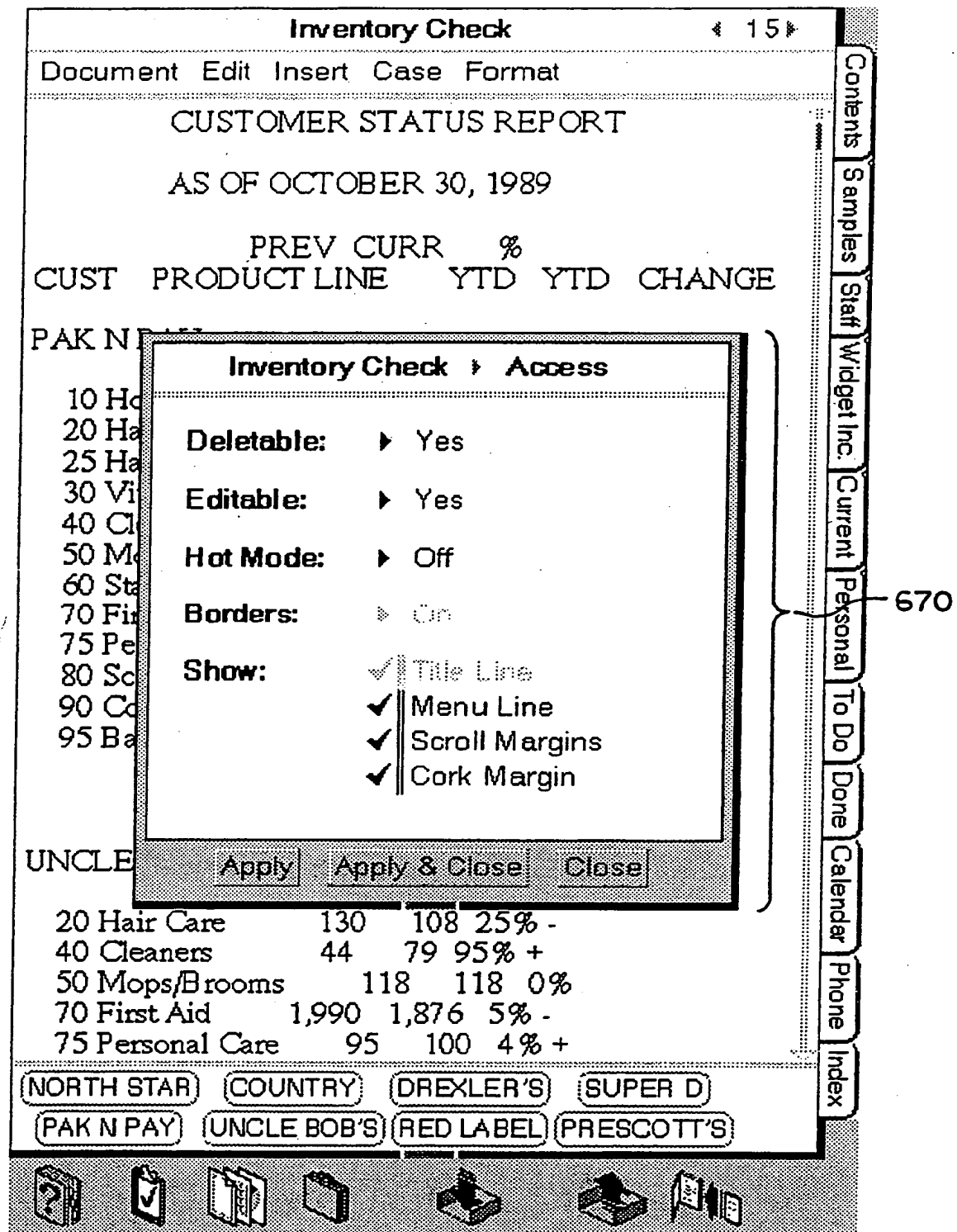
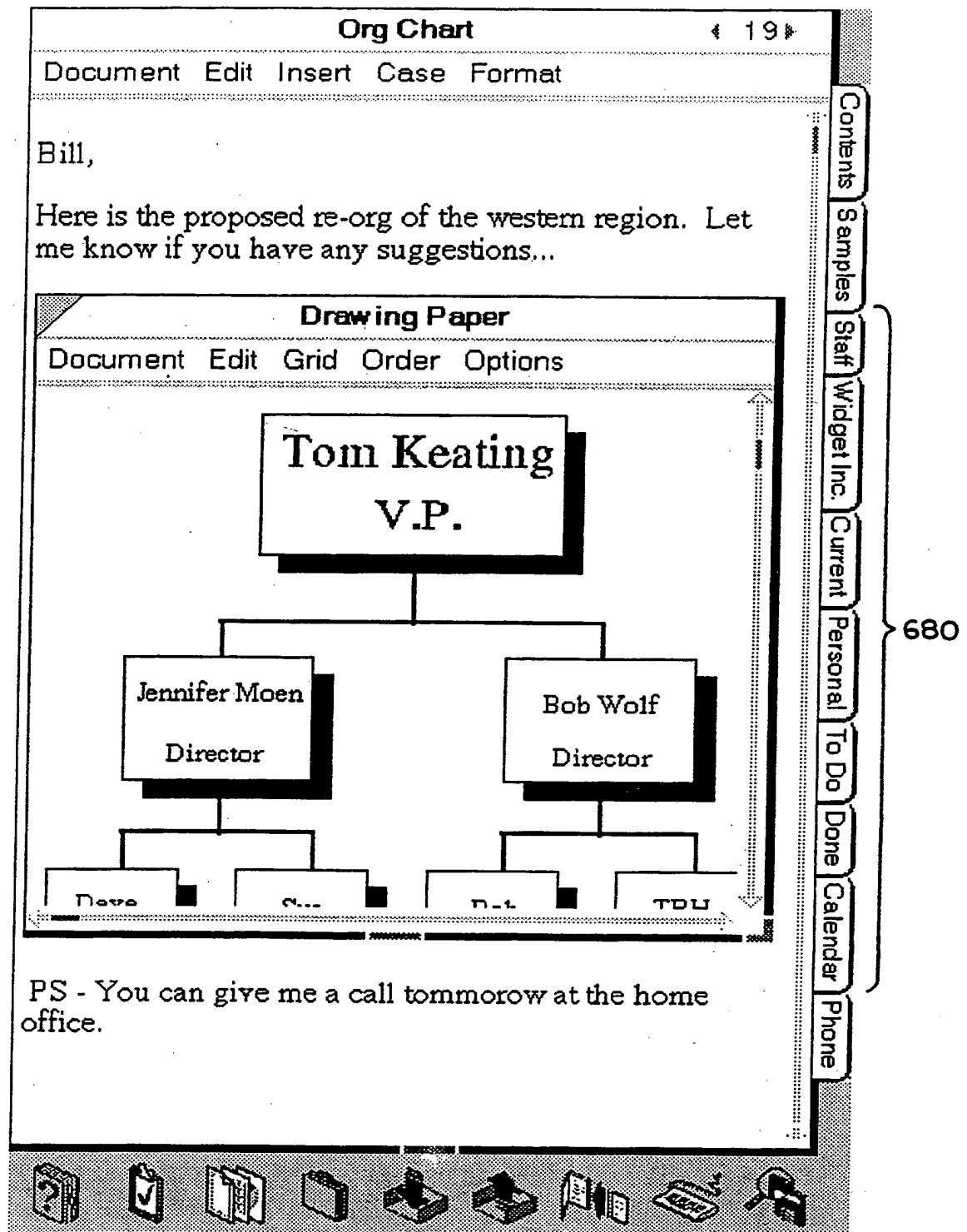


Fig. 14

SUBSTITUTE SHEET

15/48



SUBSTITUTE SHEET

16/48

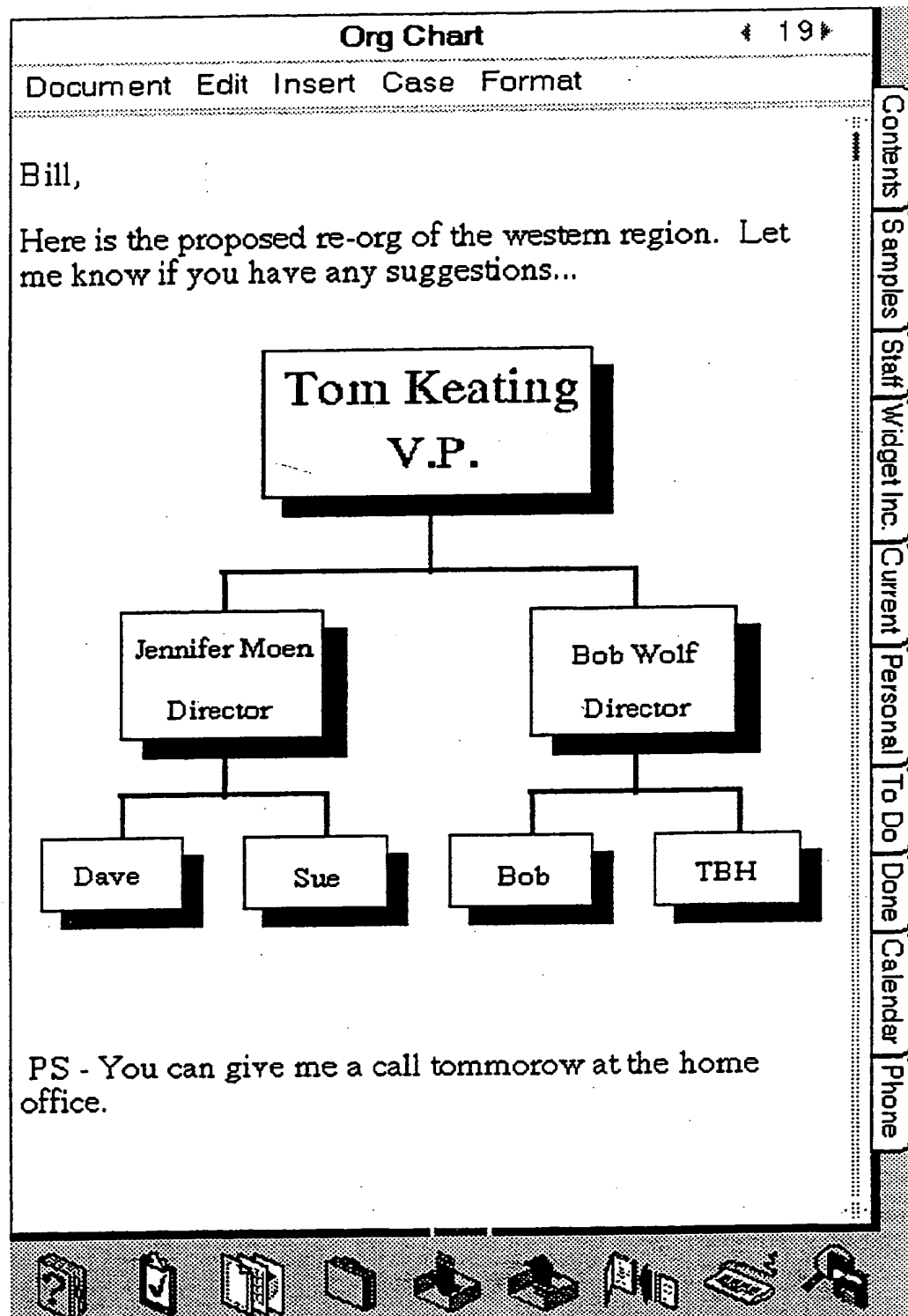


Fig. 16

SUBSTITUTE SHEET

17/48

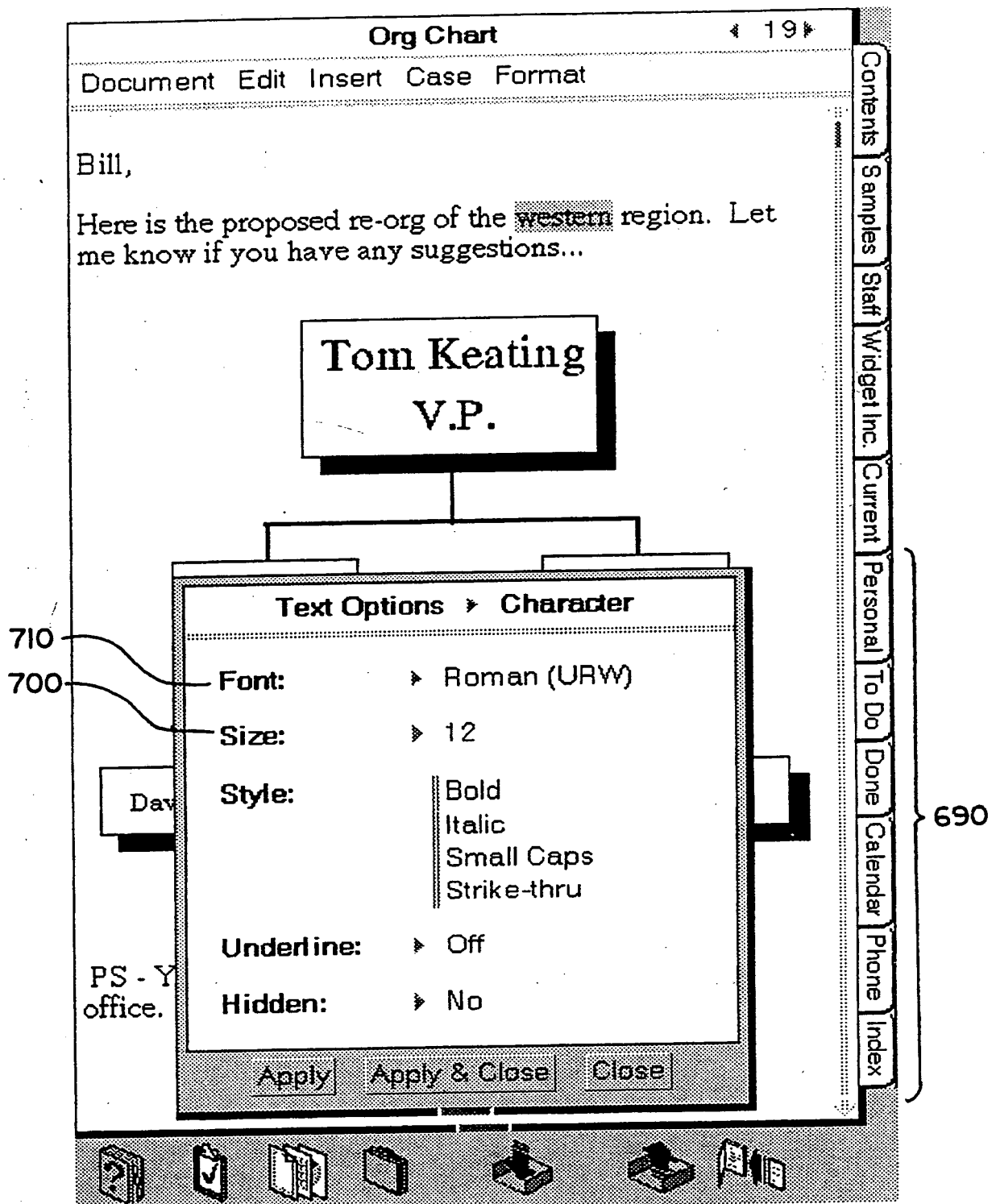


Fig. 17

SUBSTITUTE SHEET

18/48

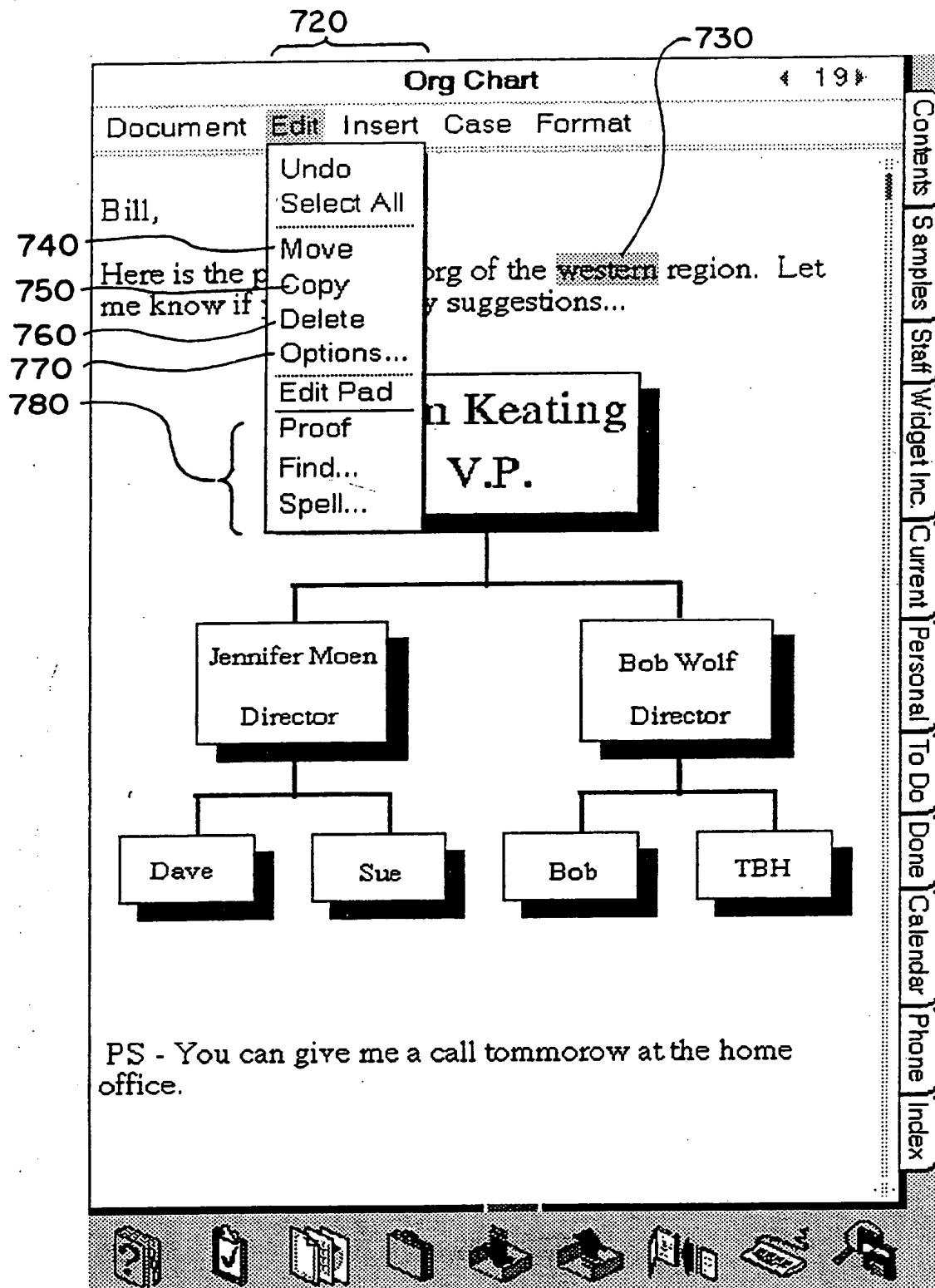
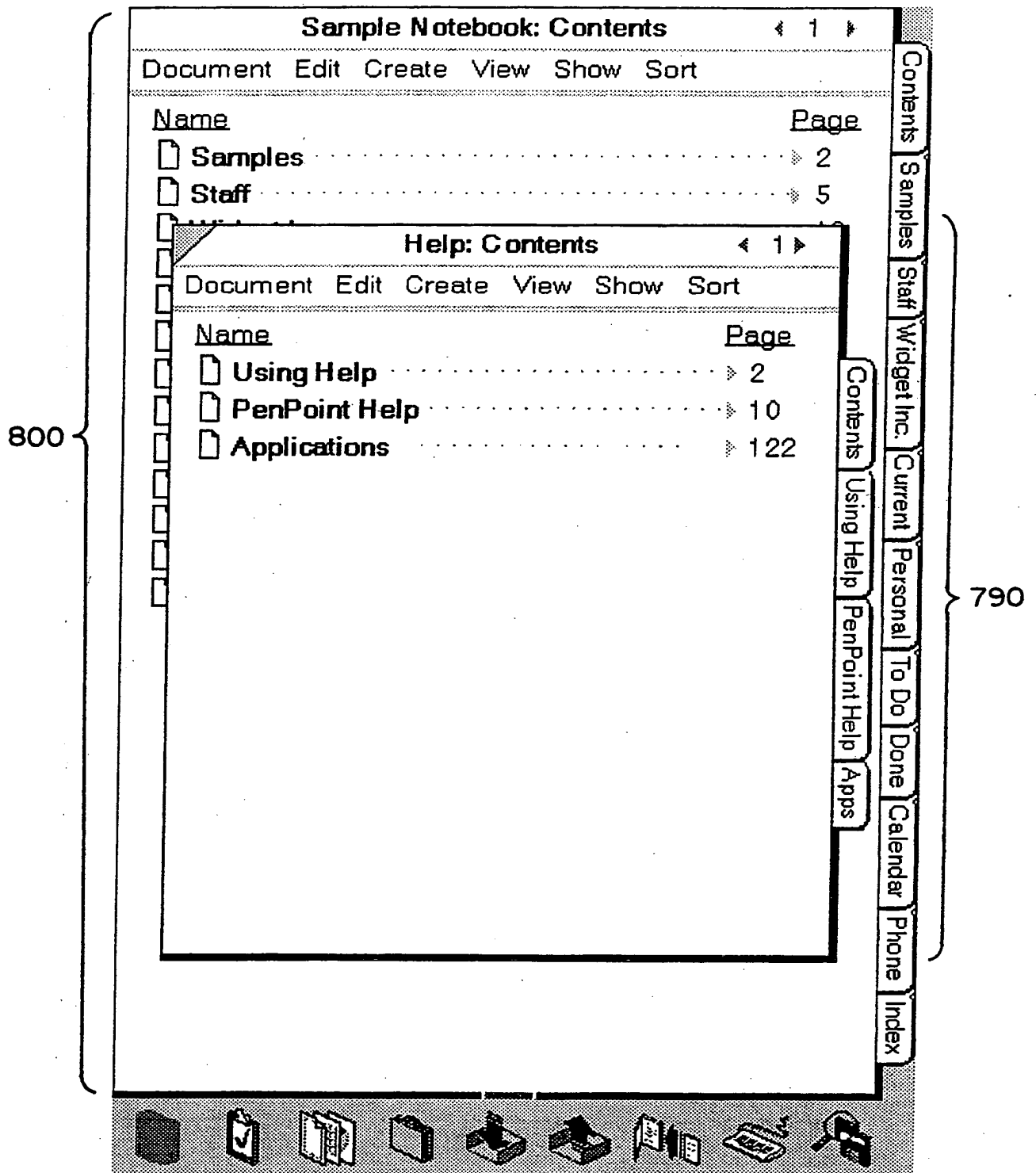


Fig. 18

SUBSTITUTE SHEET

19/48



SUBSTITUTE SHEET

20/48

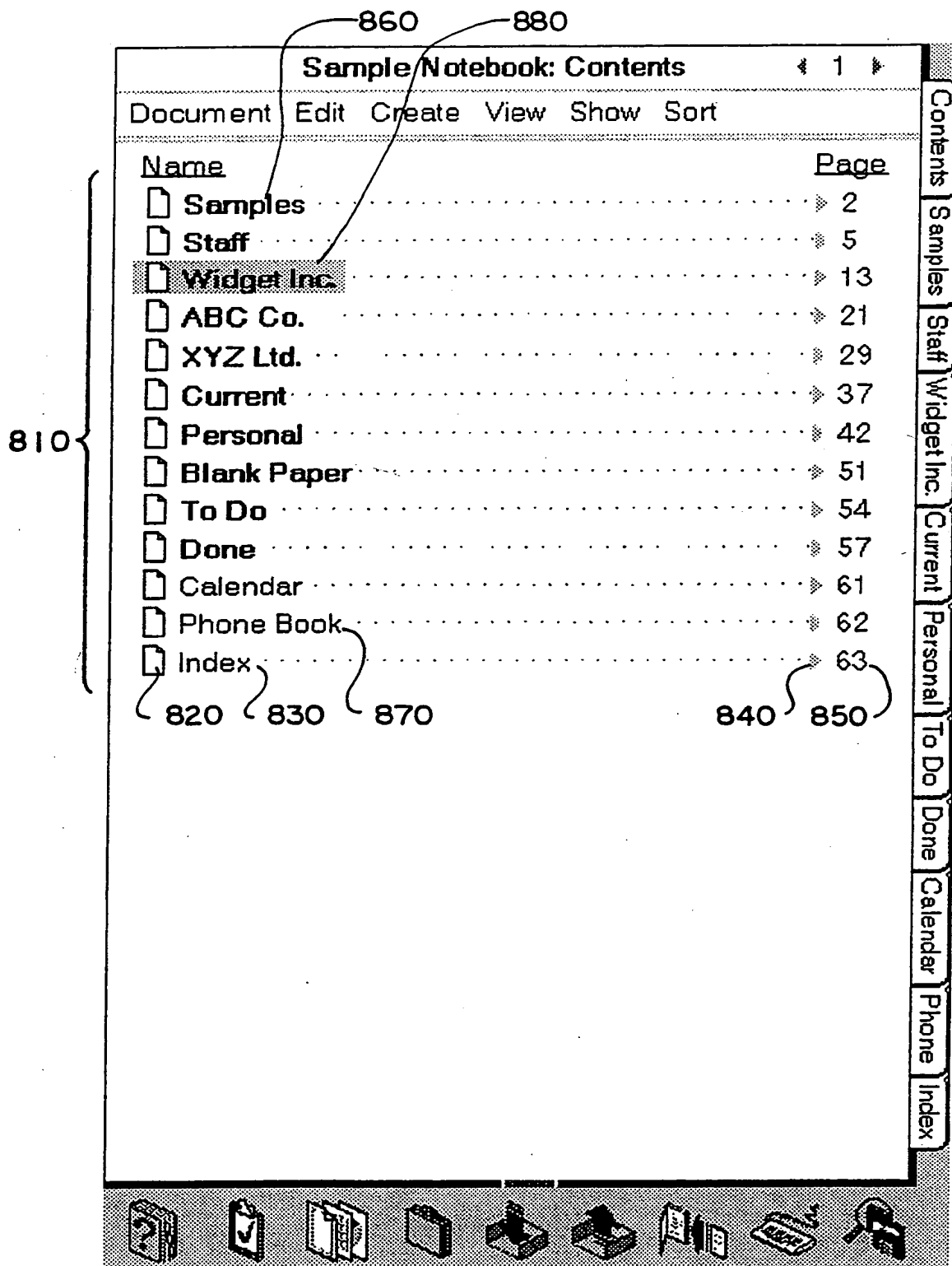


Fig. 20

SUBSTITUTE SHEET

21/48

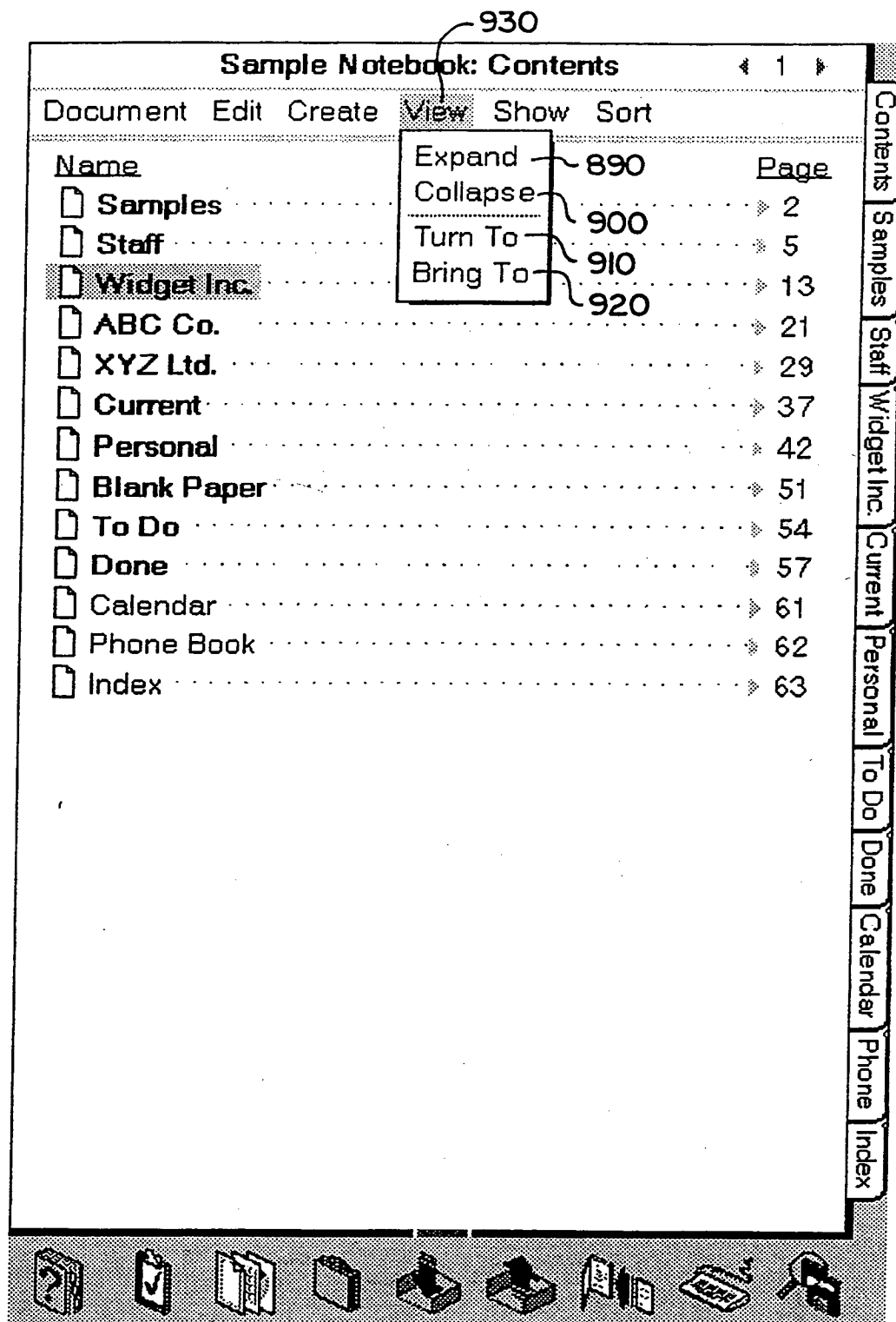


Fig. 21

SUBSTITUTE SHEET

22/48

Sample Notebook: Contents		1
Document Edit Create View Show Sort		
Name		Page
Samples		2
Staff		5
Widget Inc.		13
Contact Report		14
Inventory Check		15
Fax		16
Johnson Letter		17
New Office Layout		18
Org Chart		19
Interview Form		20
ABC Co.		21
XYZ Ltd.		29
Current		37
Personal		42
Blank Paper		51
To Do		54
Done		57
Calendar		61
Phone Book		62
Index		63

940 {

Contents Samples Staff Widget Inc. Current Personal To Do Done Calendar Phone Index

Fig. 22

SUBSTITUTE SHEET

23/48

Sample Notebook: Contents

1

Document Edit Create View Show Sort

Name	Page
Samples	2
New Product Ideas	3
Package Design Letter	4
Staff	5
Staff Meeting Notes	6
Chris	7
Pat	8
Alex	9
George	10
Follow-up: C. Dunn	11
Project Plan	12
Widget Inc.	13
Contact Report	14
Inventory Check	15
Fax	16
Johnson Letter	17
New Office Layout	18
Org Chart	19
Interview Form	20
ABC Co.	21
Contact Report	22
Inventory Check	23
Fax	24
Follow-up Letter	25
New Office Layout	26
Org Chart	27

Contents Samples Staff Widget Inc. Current Personal To Do Done Calendar Phone Index

950

Fig. 23

SUBSTITUTE SHEET

24/48

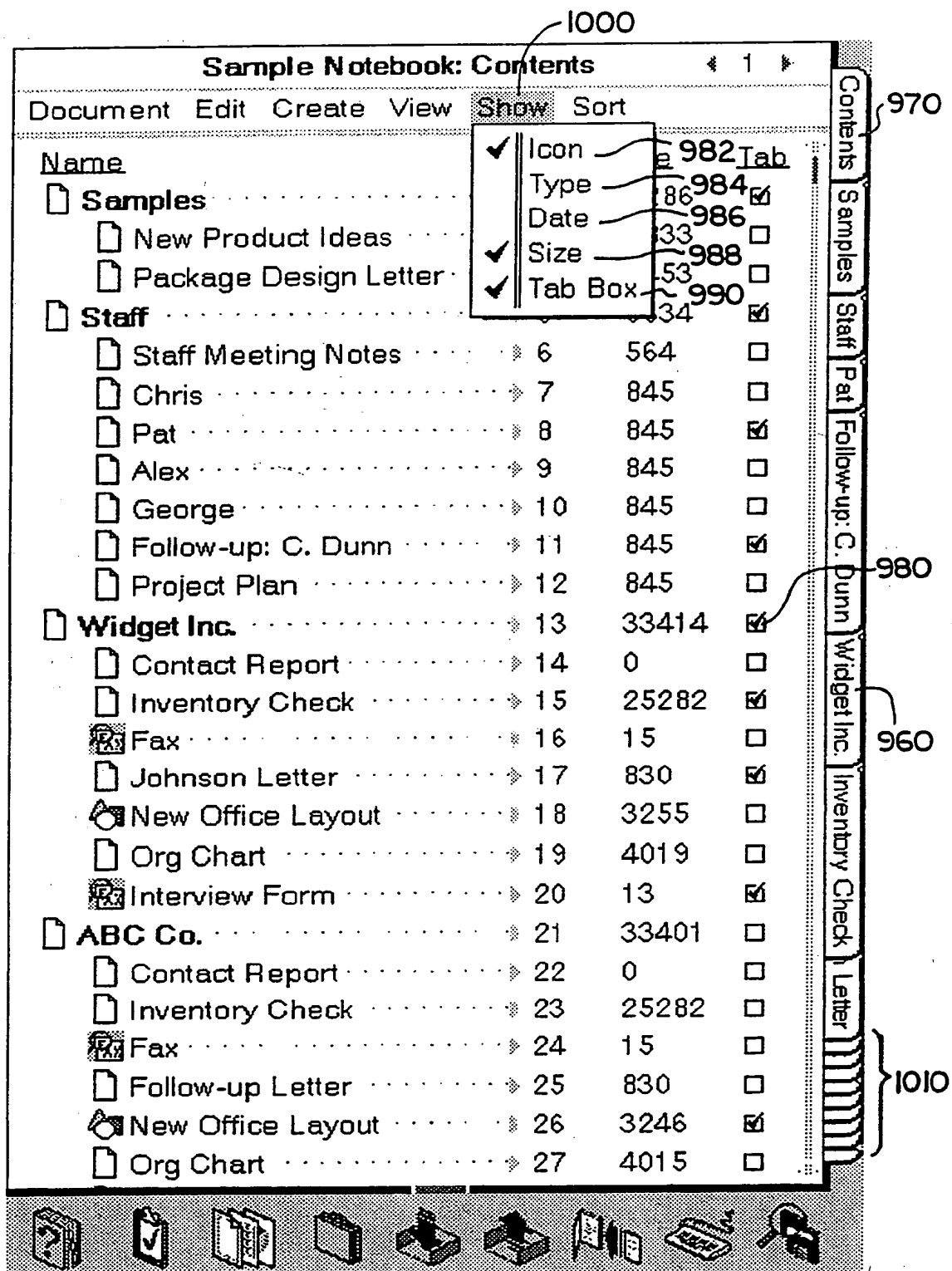


Fig. 24

SUBSTITUTE SHEET

25/48

1020

Sample Notebook: Contents			Contents	Widget Inc.	Current	Phone
Document Edit Create View Show Sort						
Name	Page	Size				
Samples	2	8286	Contents	Widget Inc.	Current	Phone
New Product Ideas	3	4833				
Package Design Letter	4	3453				
Staff	5	5634	Samples	Inventory Check	Personal	Index
Staff Meeting Notes	6	564				
Chris	7	845				
Pat	8	845				
Alex	9	845				
George	10	845				
Follow-up: C. Dunn	11	845				
Project Plan	12	845	Staff	Johnson Letter	To Do	
Widget Inc.	13	33414				
Contact Report	14	0				
Inventory Check	15	25282	Pat	Interview Form	Done	
Fax	16	15				
Johnson Letter	17	830				
New Office Layout	18	3255				
Org Chart	19	4019				
Interview Form	20	13	Follow-up: C. Dunn	New Office Layout	Calendar	
ABC Co.	21	33401				
Contact Report	22	0				
Inventory Check	23	25282				
Fax	24	15				
Follow-up Letter	25	830				
New Office Layout	26	3246				
Org Chart	27	4015				

Fig. 25

SUBSTITUTE SHEET

26/48

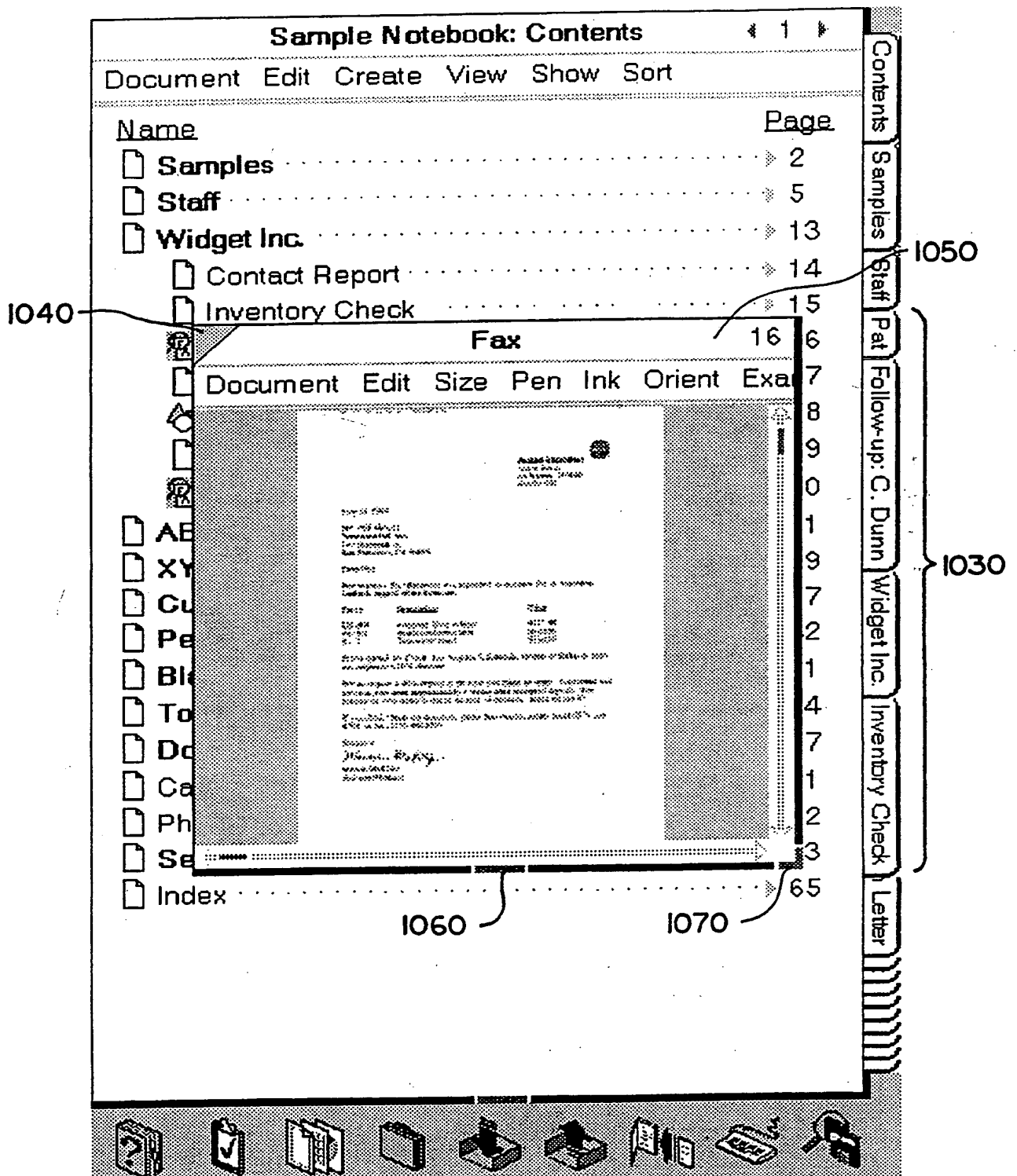
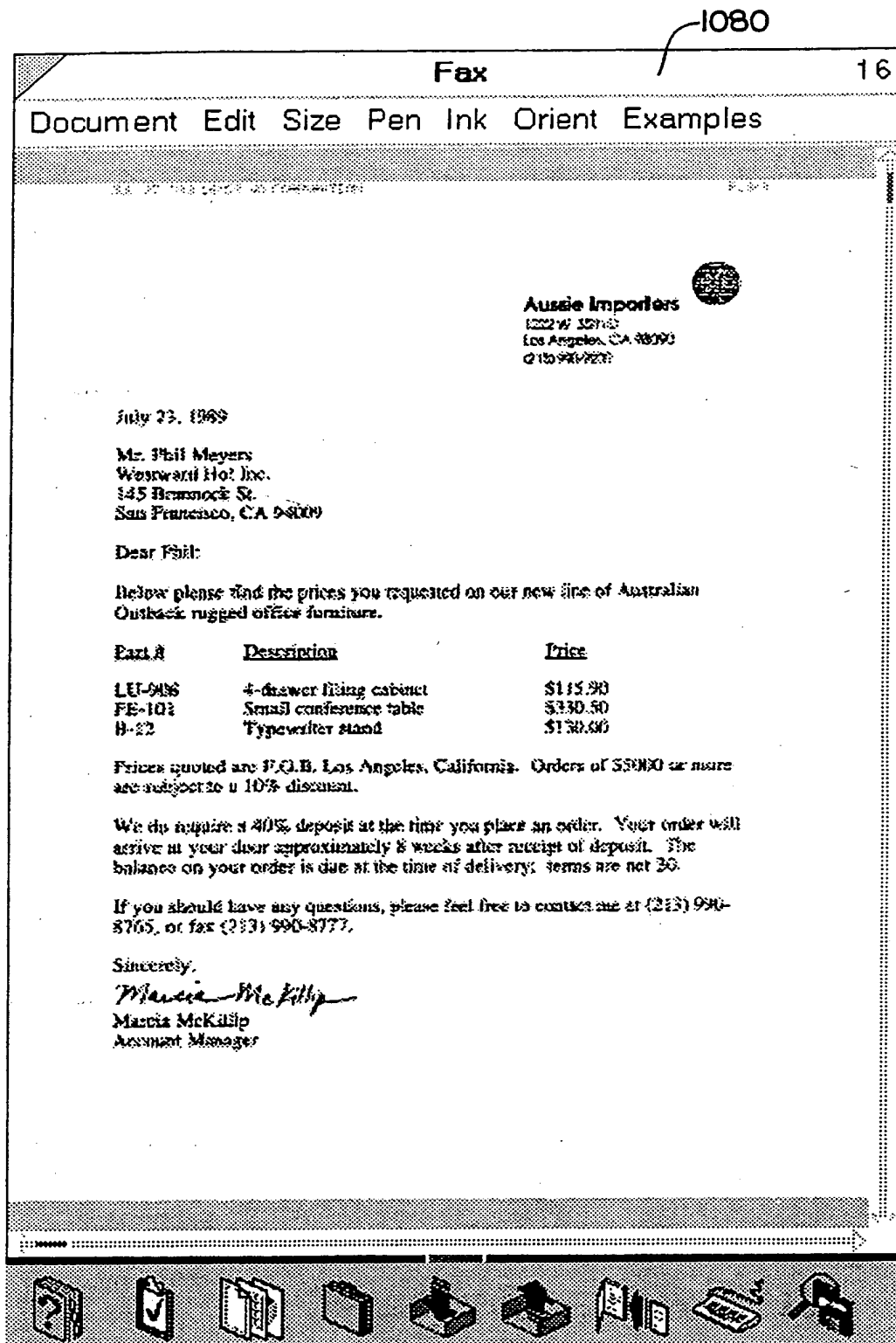


Fig. 26

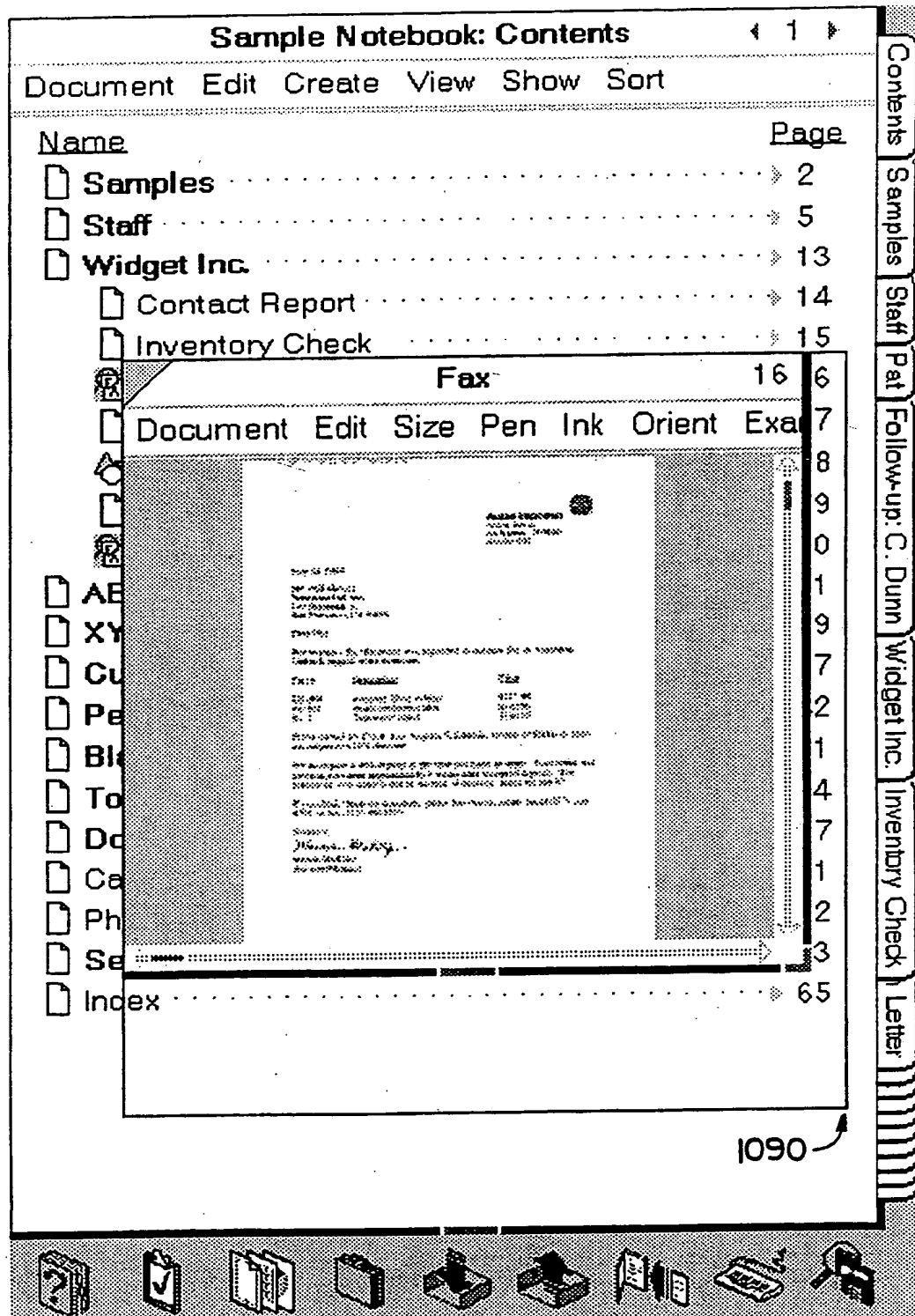
SUBSTITUTE SHEET

27/48



SUBSTITUTE SHEET

28/48



SUBSTITUTE SHEET

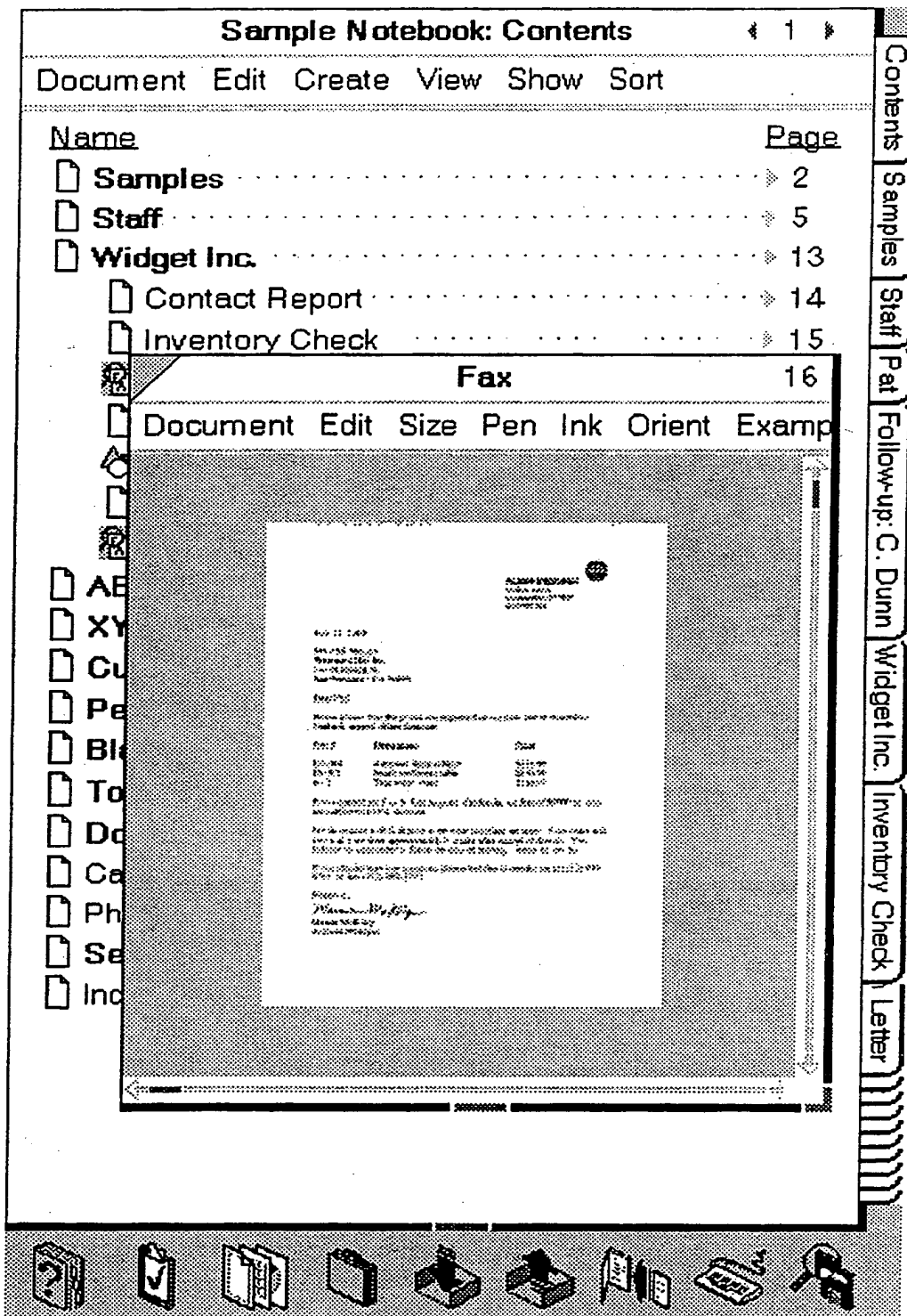


Fig. 29

SUBSTITUTE SHEET

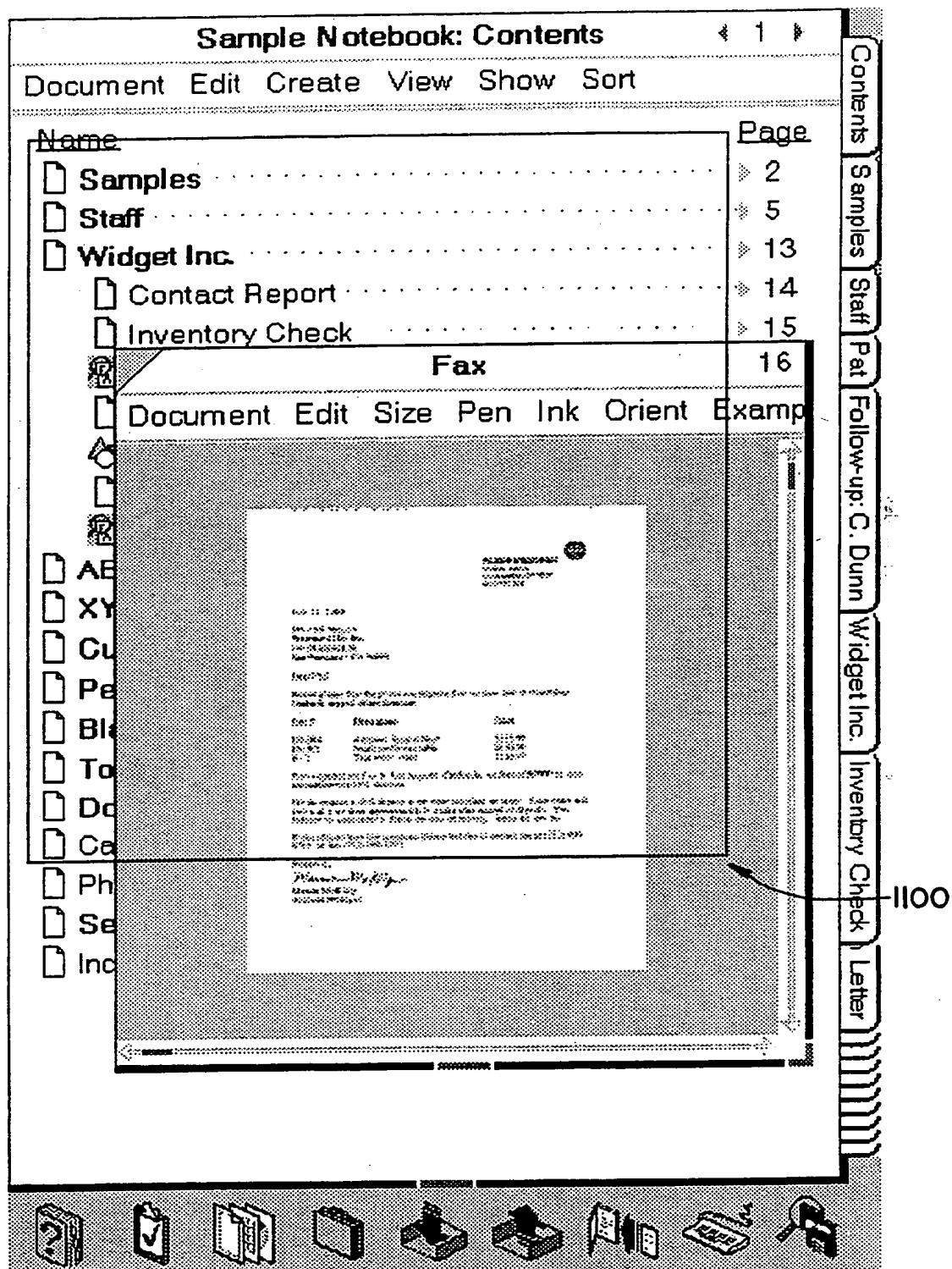


Fig. 30

SUBSTITUTE SHEET

31/48

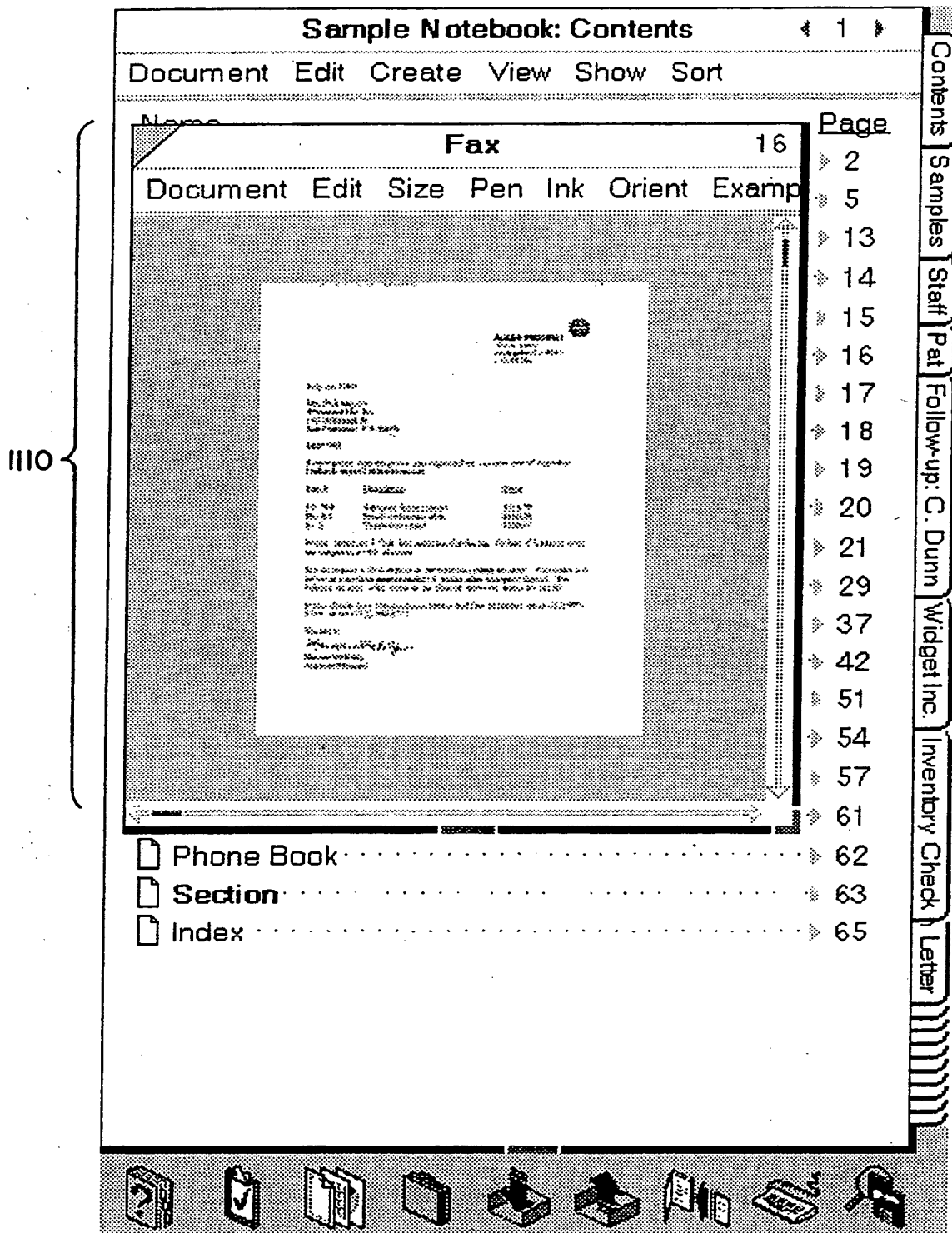


Fig. 31

SUBSTITUTE SHEET

32/48

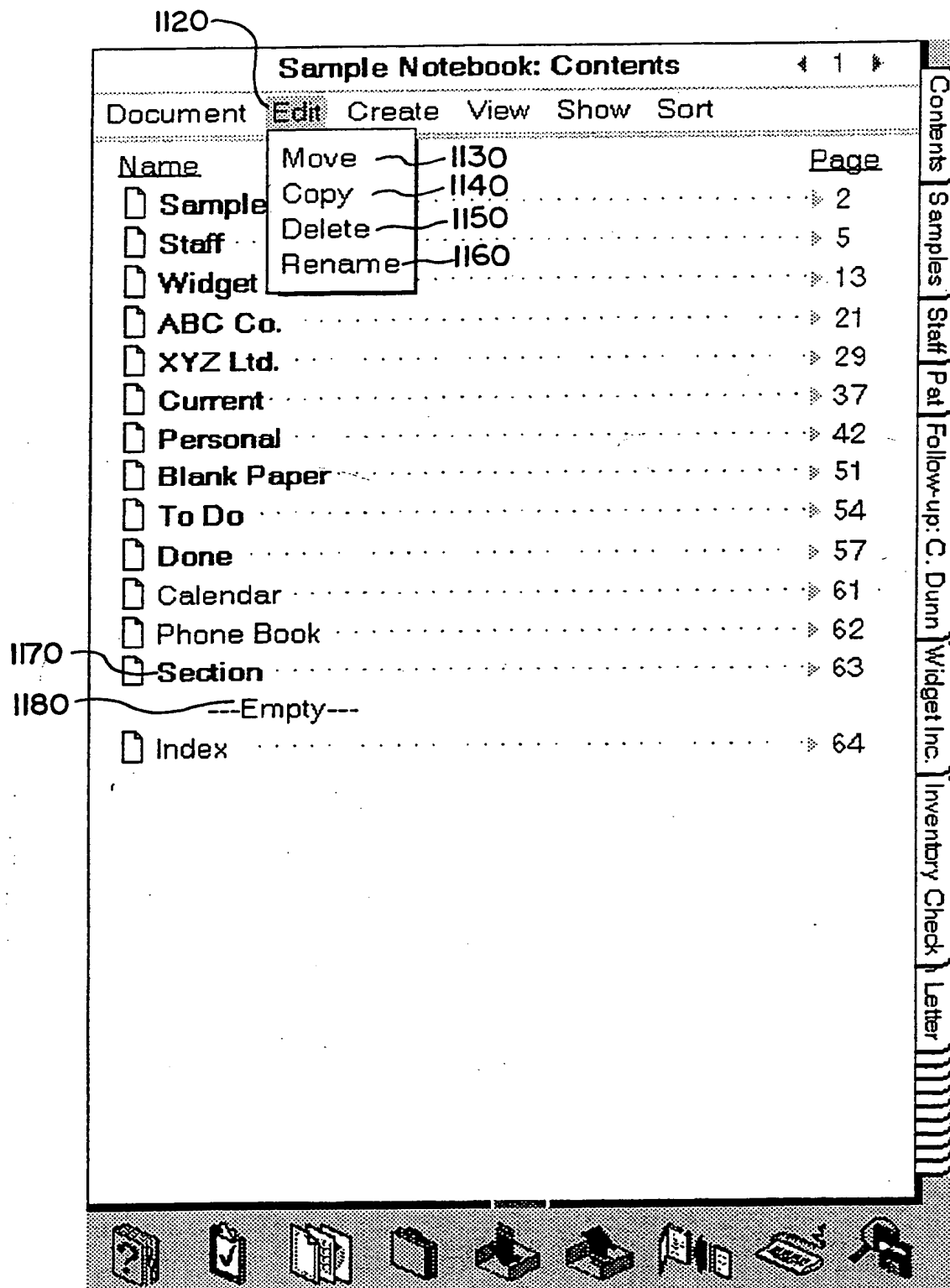


Fig. 32

SUBSTITUTE SHEET

33/48

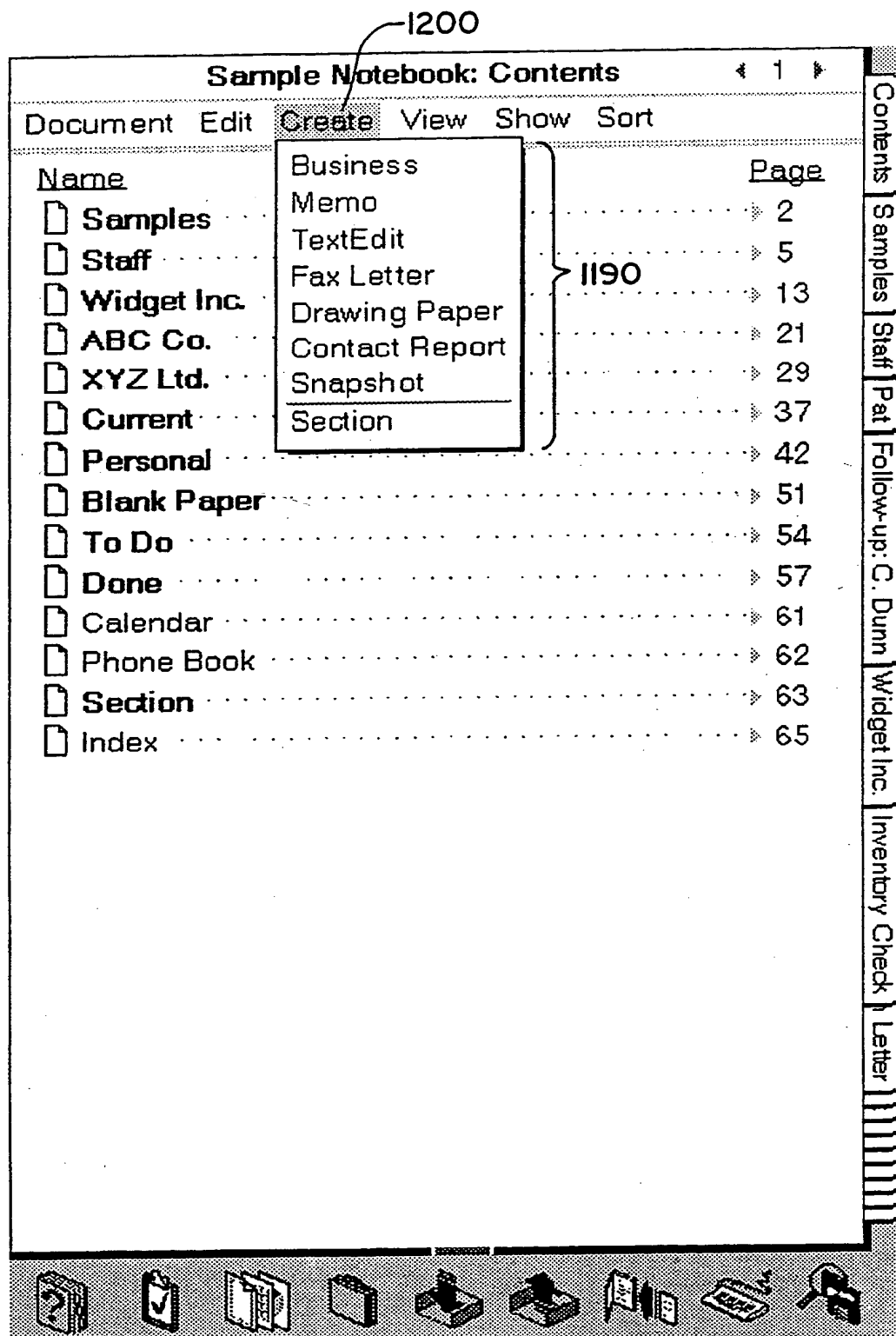


Fig. 33

SUBSTITUTE SHEET

34/48

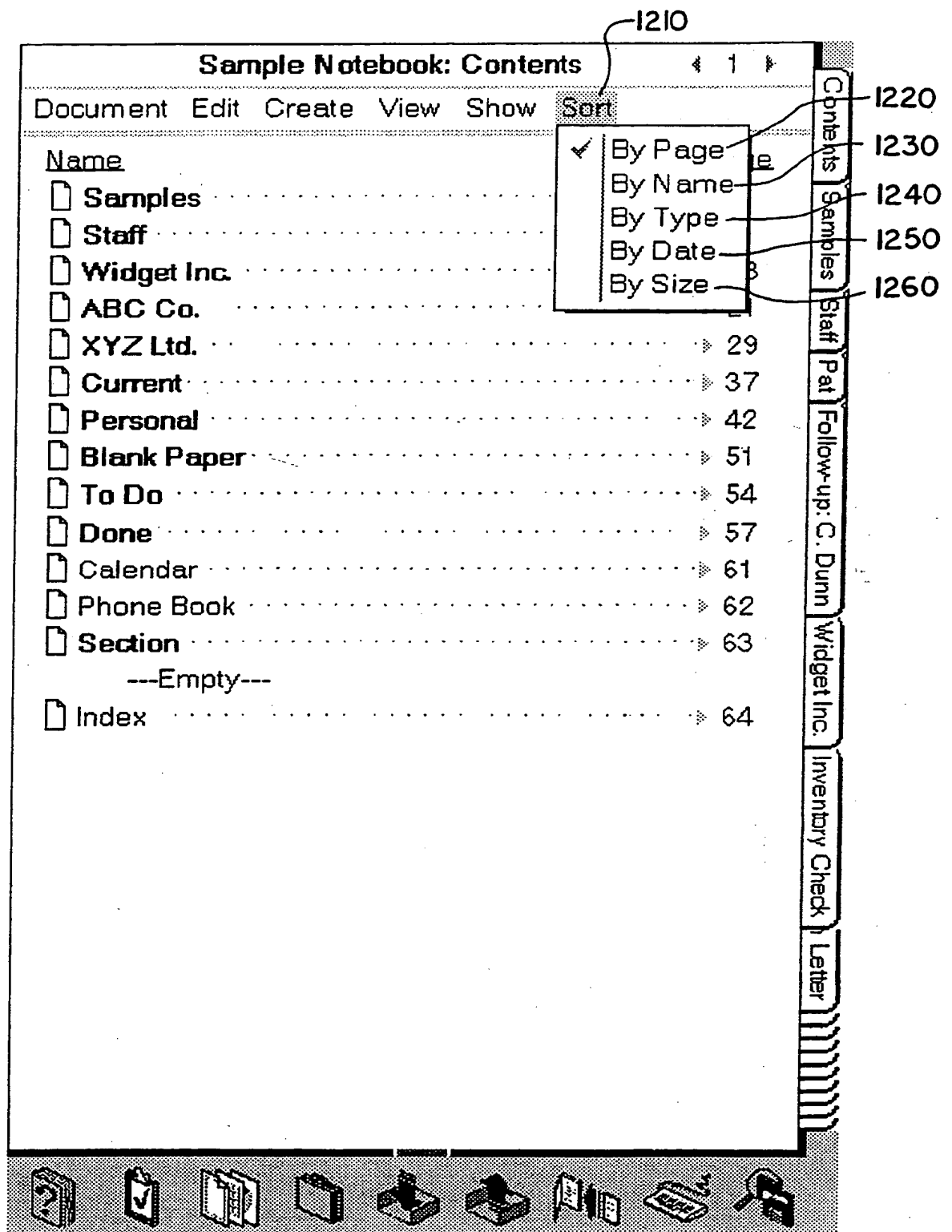
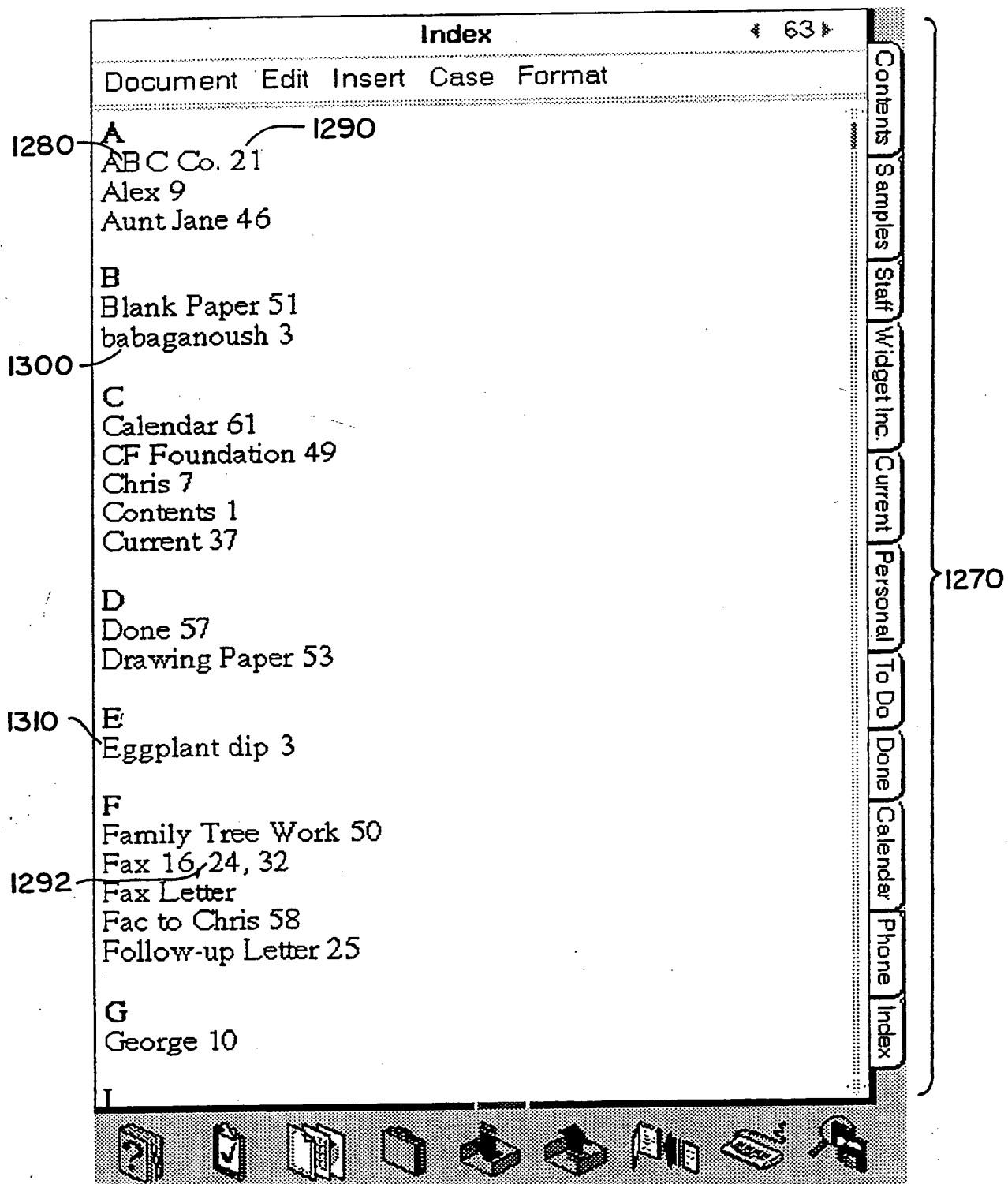


Fig. 34

SUBSTITUTE SHEET

35/48



SUBSTITUTE SHEET

36/48

Inventory Check 15

Document Edit Insert Case Format

PRESCOTT'S

20 Hair Care	200	250	20% +
25 Hardware	1,230	1,400	10% +
30 Vitamins	54	54	0%
60 Stationery	99	80	20% -
70 First Aid	34	34	0%
75 Personal care	130	135	2% +
90 Cosmetics	470	500	7% +
TOTAL 2,300 2,500 10% +			

NORTH STAR

40 Cleaners	130	106	25% -
50 Mops/Brooms	44	78	90% +
60 Stationery	120	120	0%
70 First Aid	1,990	1,876	5% -
75 Personal Care	90	95	4% +
TOTAL 2,400 2,600 8% +			

U-PAK-IT DRUGS

40 Cleaners	130	106	25% -
50 Mops/Brooms	44	78	90% +
70 First Aid	1,990	1,876	5% -
75 Personal Care	90	95	4% +
TOTAL 2,350 2,290 5% -			

NORTH STAR
COUNTRY
DREXLER'S
SUPER D

PAK N PAY
UNCLE BOB'S
RED LABEL
PRESCOTT'S




















Fig. 36

1320

SUBSTITUTE SHEET

37/48

1350

Inventory Check
15

Document
Edit
Insert
Case
Format

PAK N PAY

See also SUPER D

10 Housewares	1,240	1,800	30% +
20 Hair Care	670	640	7% -
25 Hardware	8,009	5,340	40% +
30 Vitamins	400	590	20% +
40 Cleaners	130	106	25% -
50 Mops/Brooms	44	78	90% +
60 Stationery	120	120	0%
70 First Aid	1,990	1,876	5% -
75 Personal Care	90	95	4% +
80 School Supplies	439	440	0%
90 Cosmetics	440	445	1% +
95 Baby Care	29	55	88% +
TOTAL 12,008 12,987 8% +			

UNCLE BOB'S DRUGS

20 Hair Care	130	108	25% -
40 Cleaners	44	79	95% +
50 Mops/Brooms	118	118	0%
70 First Aid	1,990	1,876	5% -
75 Personal Care	95	100	4% +
TOTAL 2,400 2,600 8% +			

RED LABEL STORES

90 Cosmetics	105	100	5% -
95 Baby Care	40	75	95% +

NORTH STAR
COUNTRY
DREXLER'S
SUPER D

PAK N PAY
UNCLE BOB'S
RED LABEL
PRESCOTT'S

Contents
Samples
Staff
Pat
Follow-up: C. Dunn
Widget Inc.
Inventory Check
Letter




















Fig. 37

SUBSTITUTE SHEET

38/48

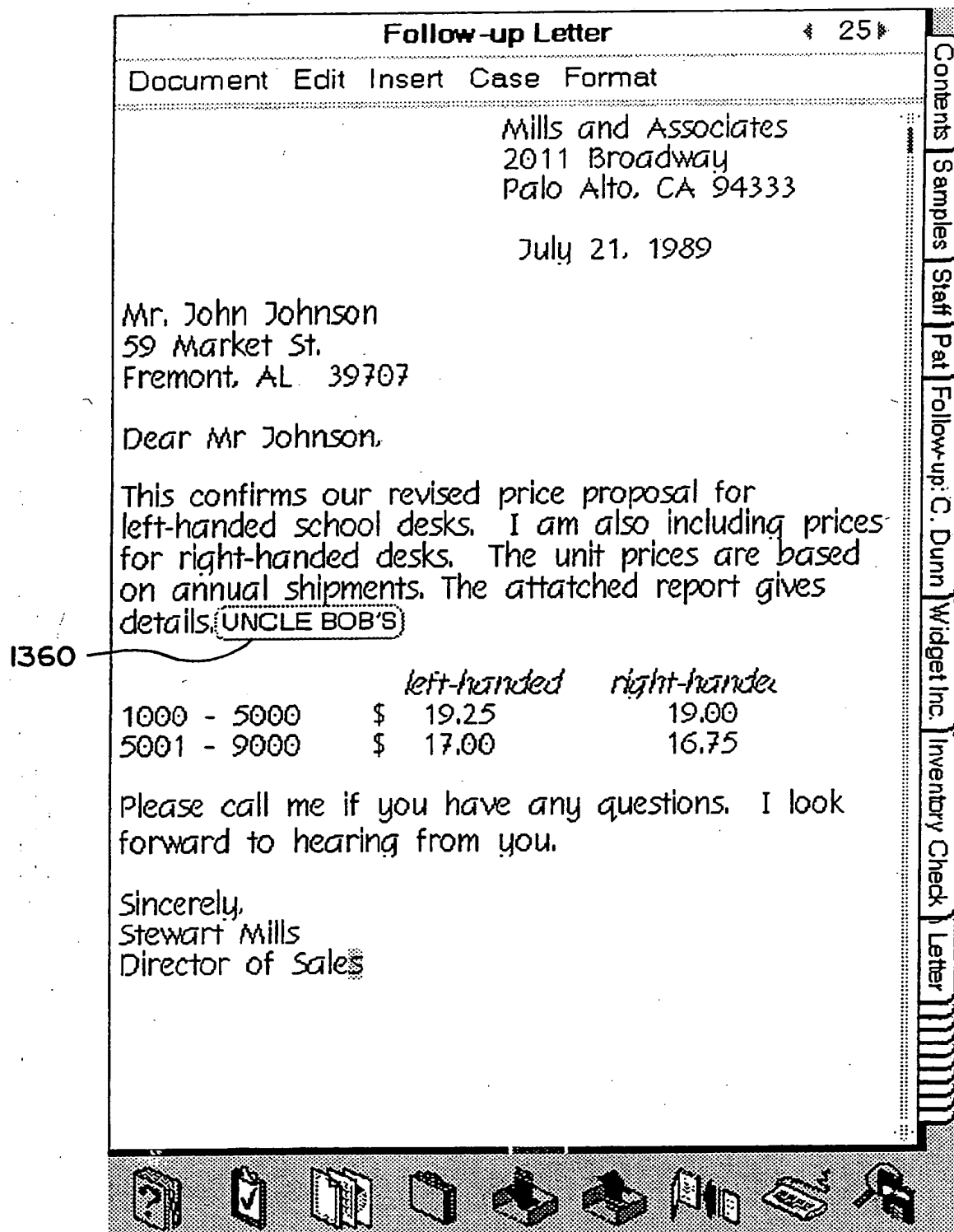


Fig. 38

SUBSTITUTE SHEET

39/48

1370 {

Follow-up Letter 25

Document Edit Insert Case Format

Mills and Associates
2011 Broadway
Palo Alto, CA 94333

July 21, 1989

Mr. John Johnson
59 Ma
Fremor

Dear Mr. Johnson,

This copy is for right-hand details.

1000
5001

Inventory Check 15

Document Edit Insert Case Format

TOTAL 12,008 12,987 8% +

UNCLE BOB'S DRUGS

20 Hair Care	130	108	25% -
40 Cleaners	44	79	95% +
50 Mops/Brooms	118	118	0%
70 First Aid	1,990	1,876	5% -
75 Personal Care	95	100	4% +
TOTAL		2,400	2,600 8% +

Please forward Sincerely,
Stewart Mills
Director of Sales

DREXLER'S

SUPER D

PRESCOTT'S

NORTH STAR

COUNTRY

PAK N PAY

UNCLE BOB'S

RED LABEL

Fig. 39

SUBSTITUTE SHEET

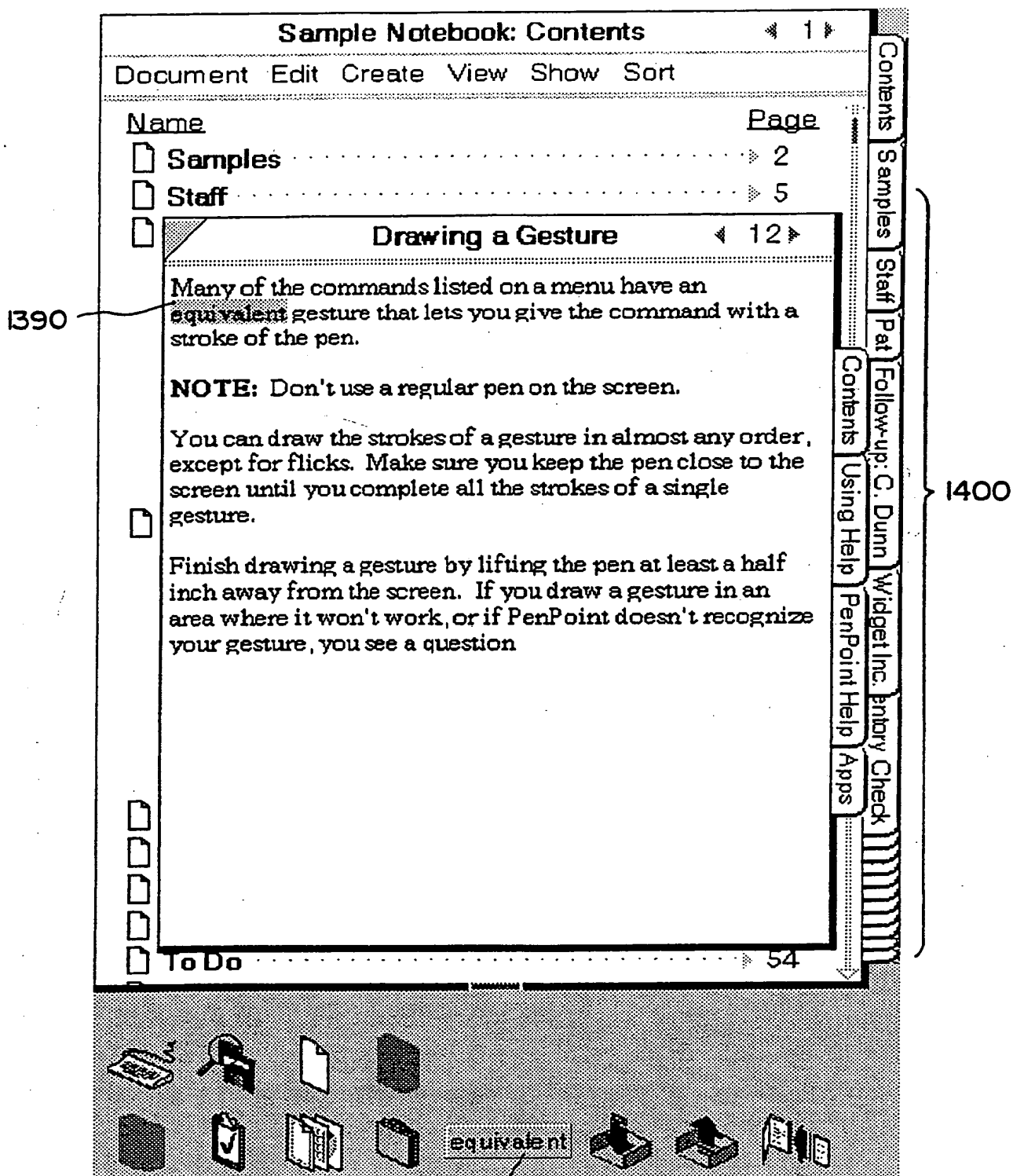


Fig. 40

41/48




























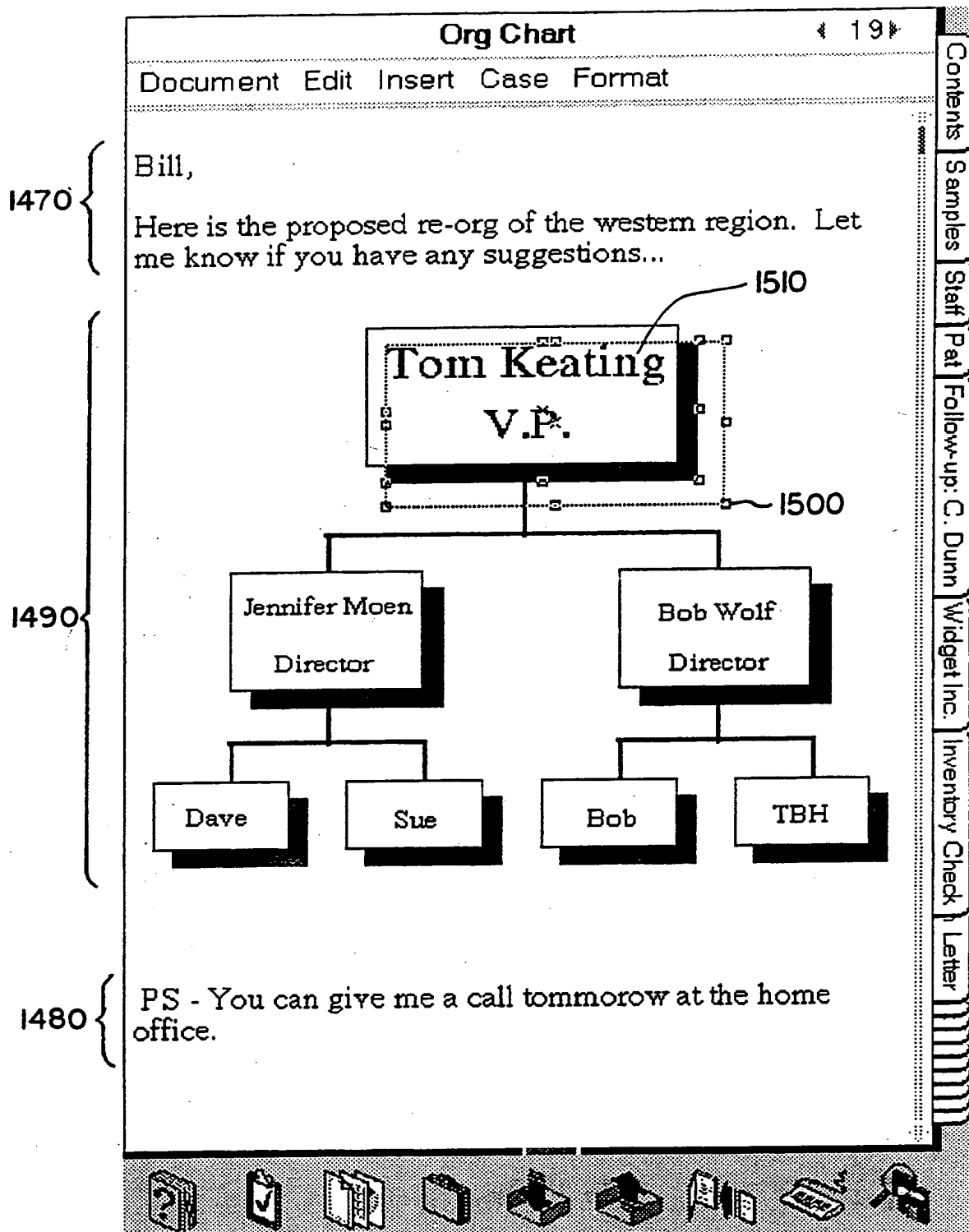
Name	Type
 PENPOINT	Dir
 SYS	Dir
1460 ~  doc	DTAPP
 Accessories	ACCESSRY
 Devices	DEVNB
 Disks	DSKVIEW
 docstate.res	File
 Help	HELPNB
 Inbox	INBOXNB
 Installer	INSTLUI
 Keyboard	Keyboard
 Outbox	OBOXNB
 Preferences	PREFS
1450 ~  Sample Notebook	Notebook
1440 ~  Contents	Section
 docstate.res	File
 Done	Section
 Index	TextEdit
 Phone Book	TextEdit
 Samples	Section
 Section	Section
1430 ~  Widget Inc.	Section
1420 ~  docstate.res	File
 Org Chart	TextEdit
1410 ~  docstate.res	File
 Drawing Paper	DrawingPap
1412 ~  docstate.res	File

Fig. 41

SUBSTITUTE SHEET



43/48

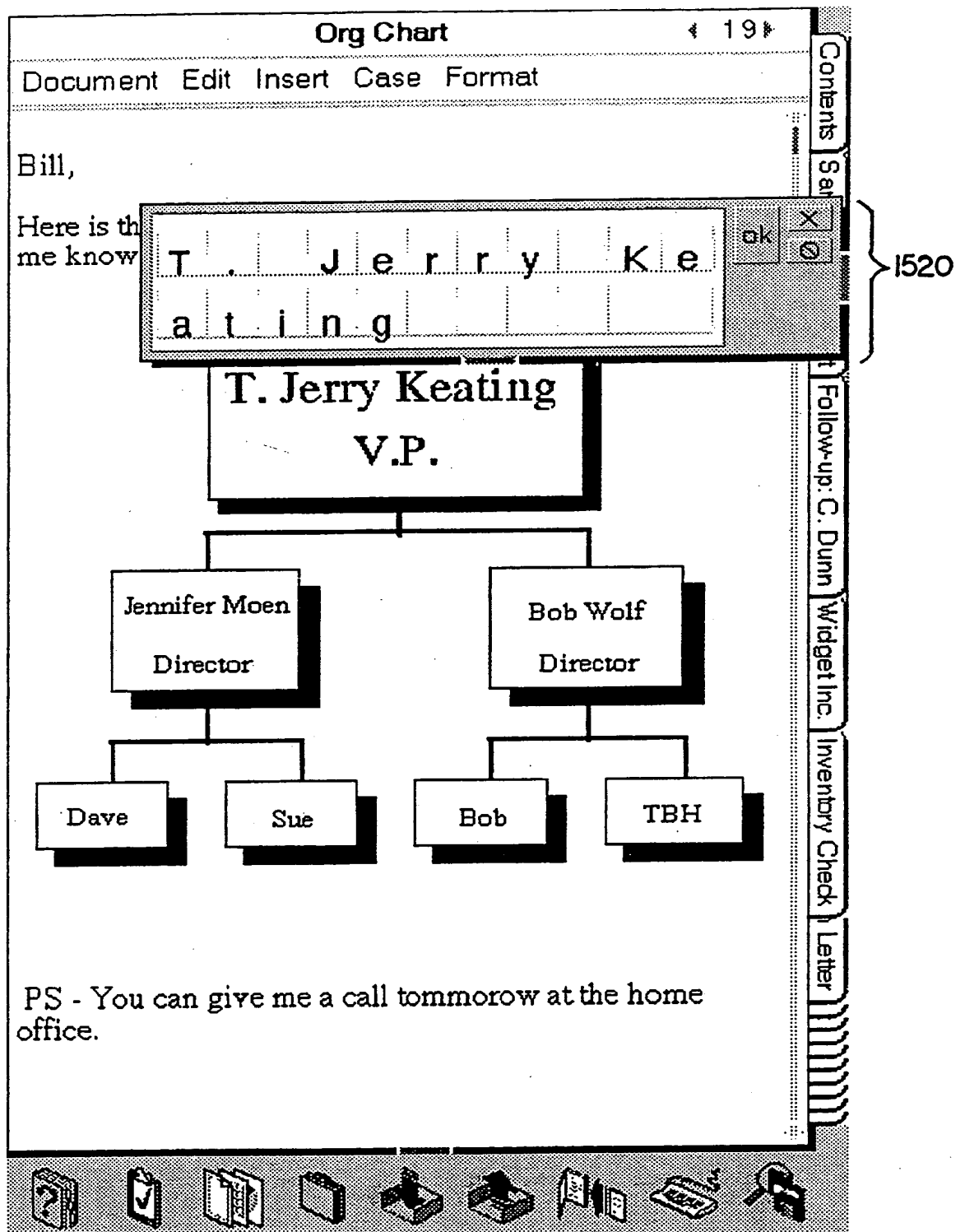


Fig. 43

SUBSTITUTE SHEET

44/48

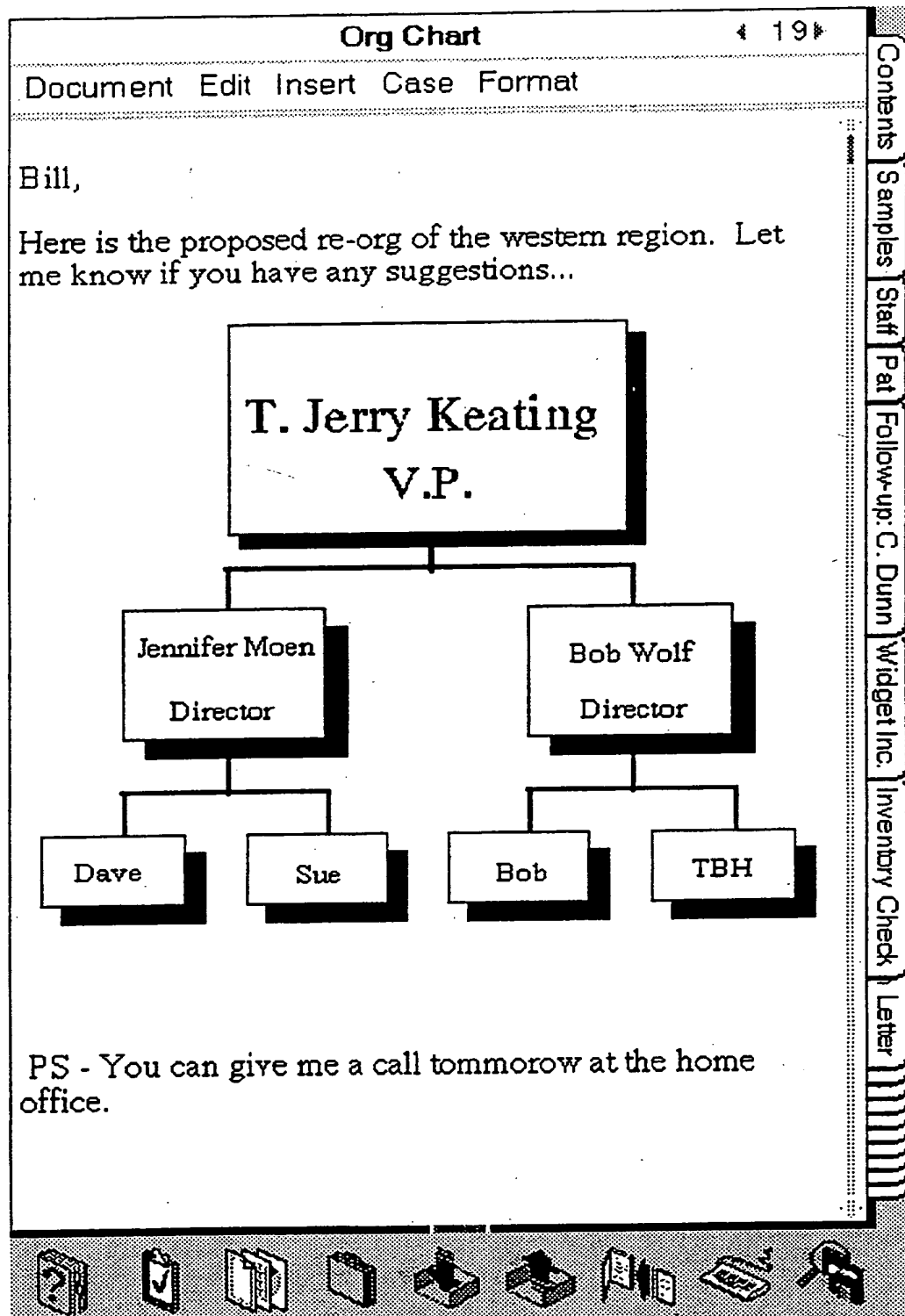


Fig. 44

SUBSTITUTE SHEET

45/48

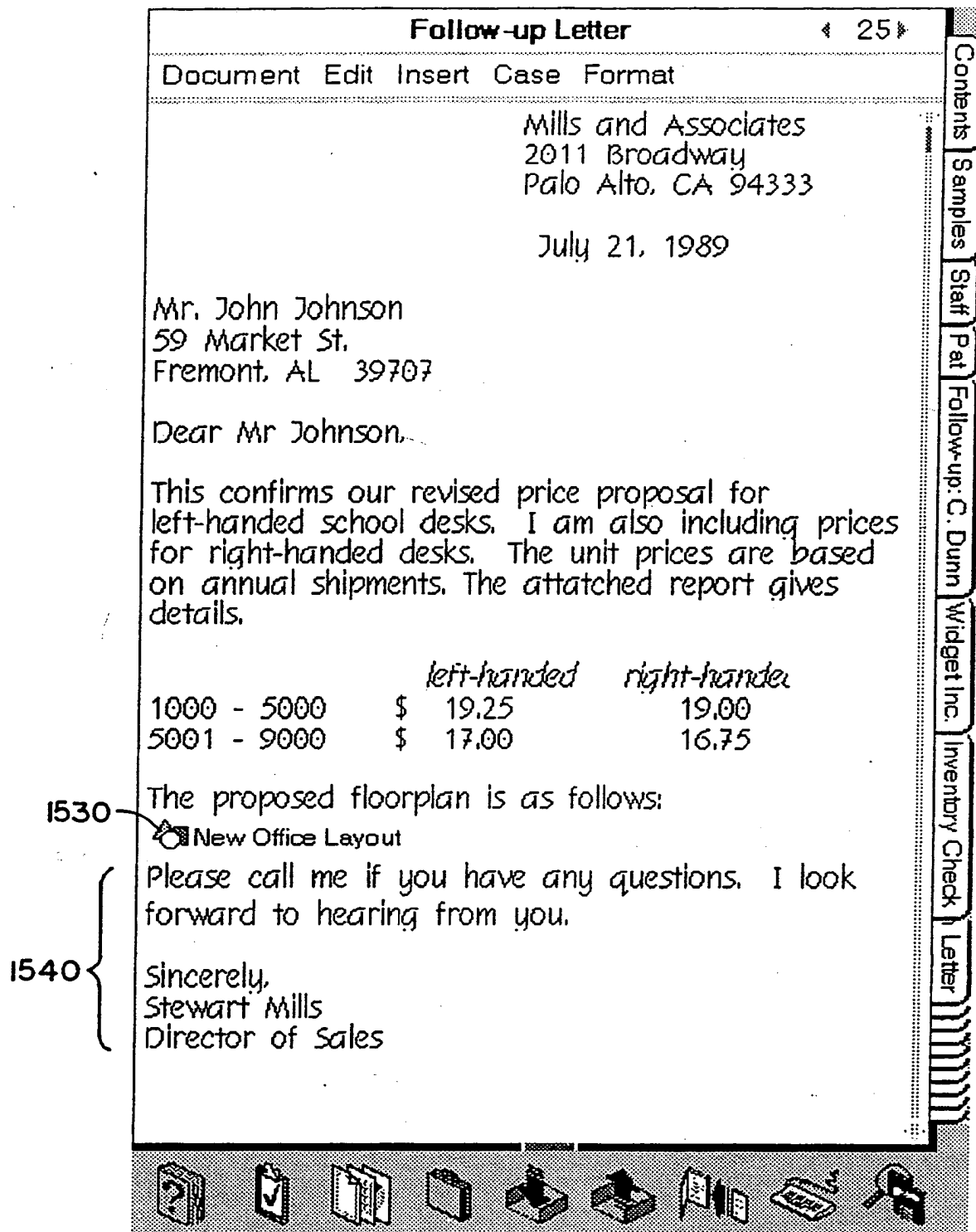


Fig. 45

SUBSTITUTE SHEET

46/48

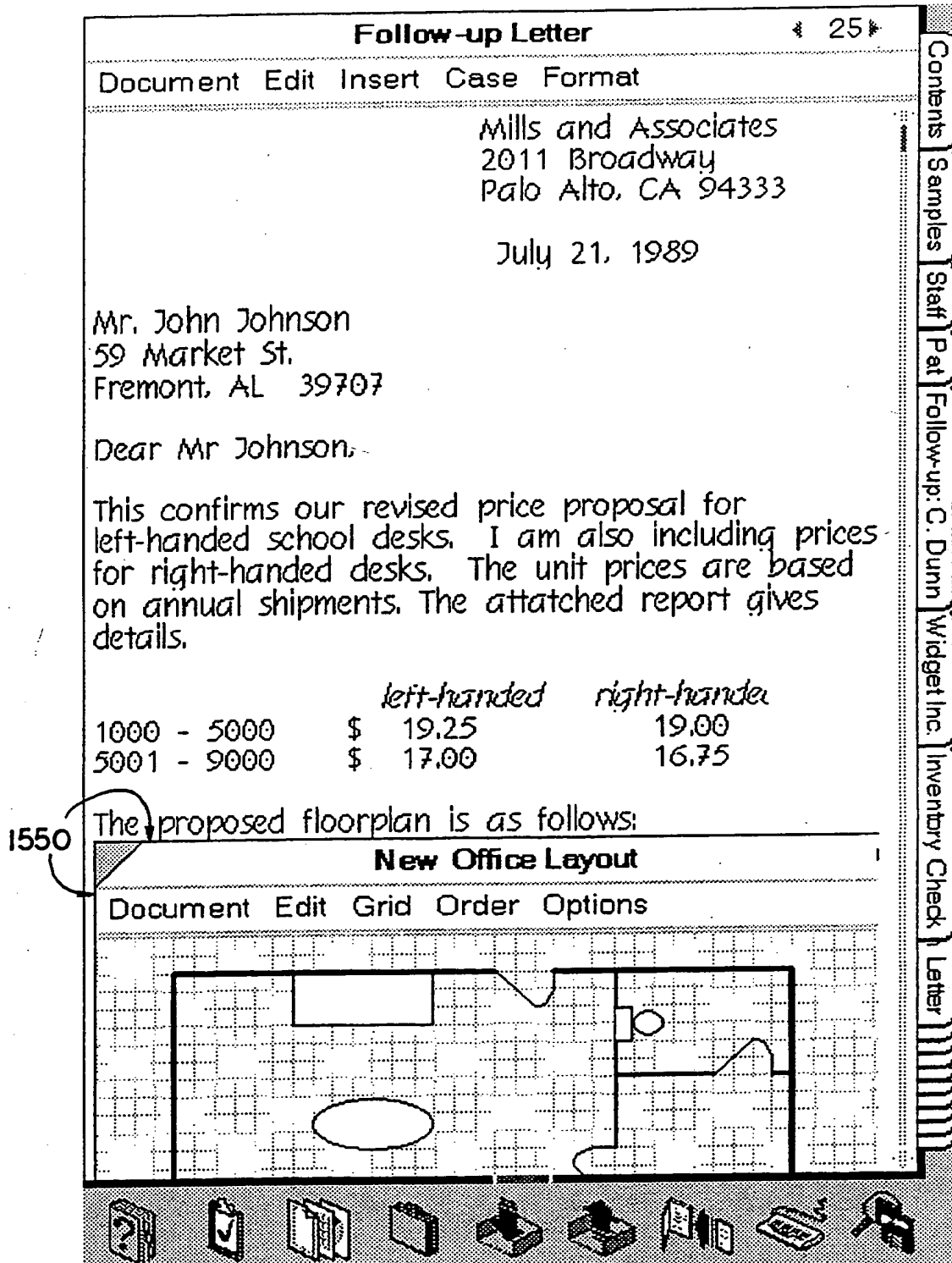


Fig. 46

SUBSTITUTE SHEET

47/48

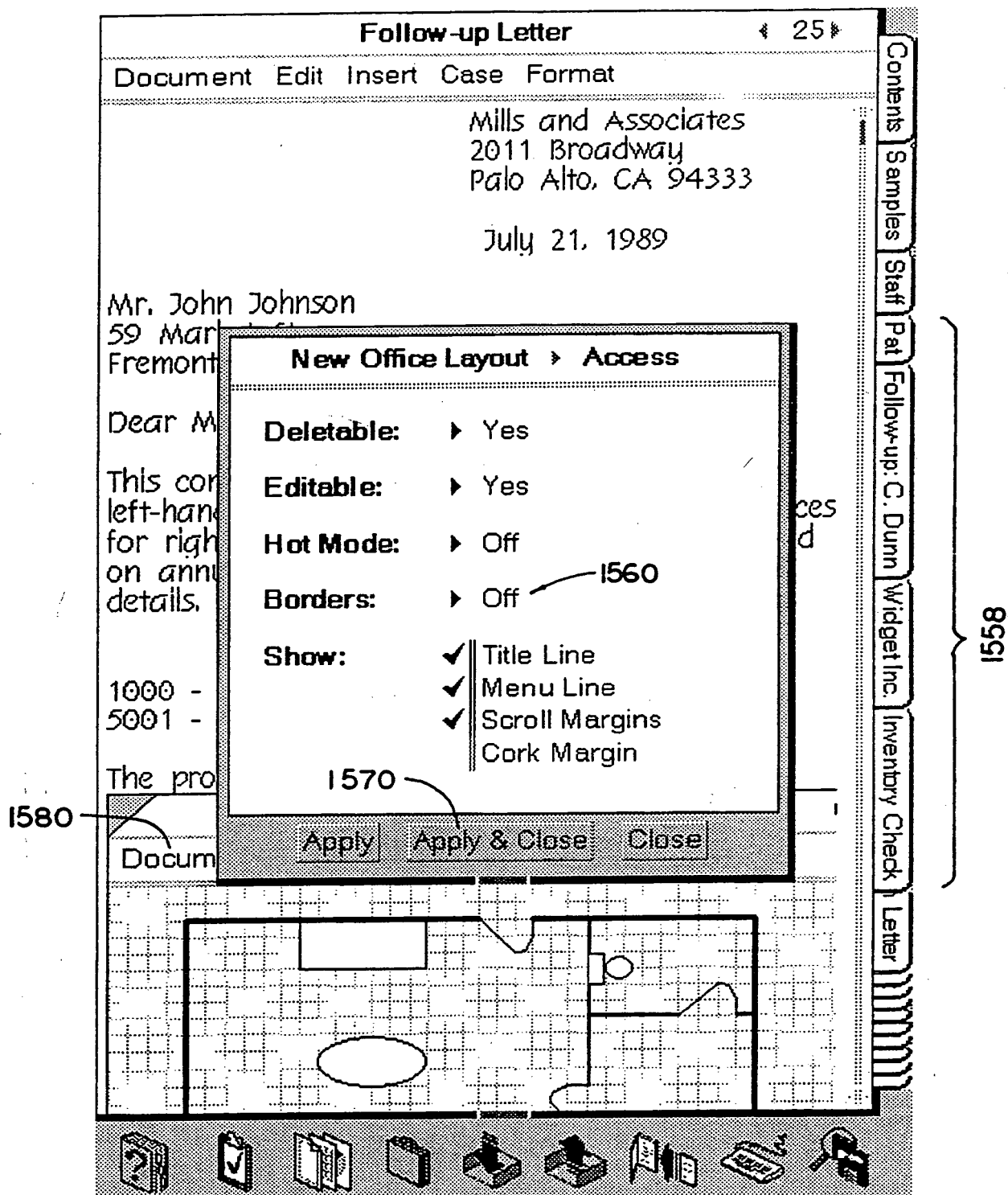


Fig. 47

SUBSTITUTE SHEET

48/48

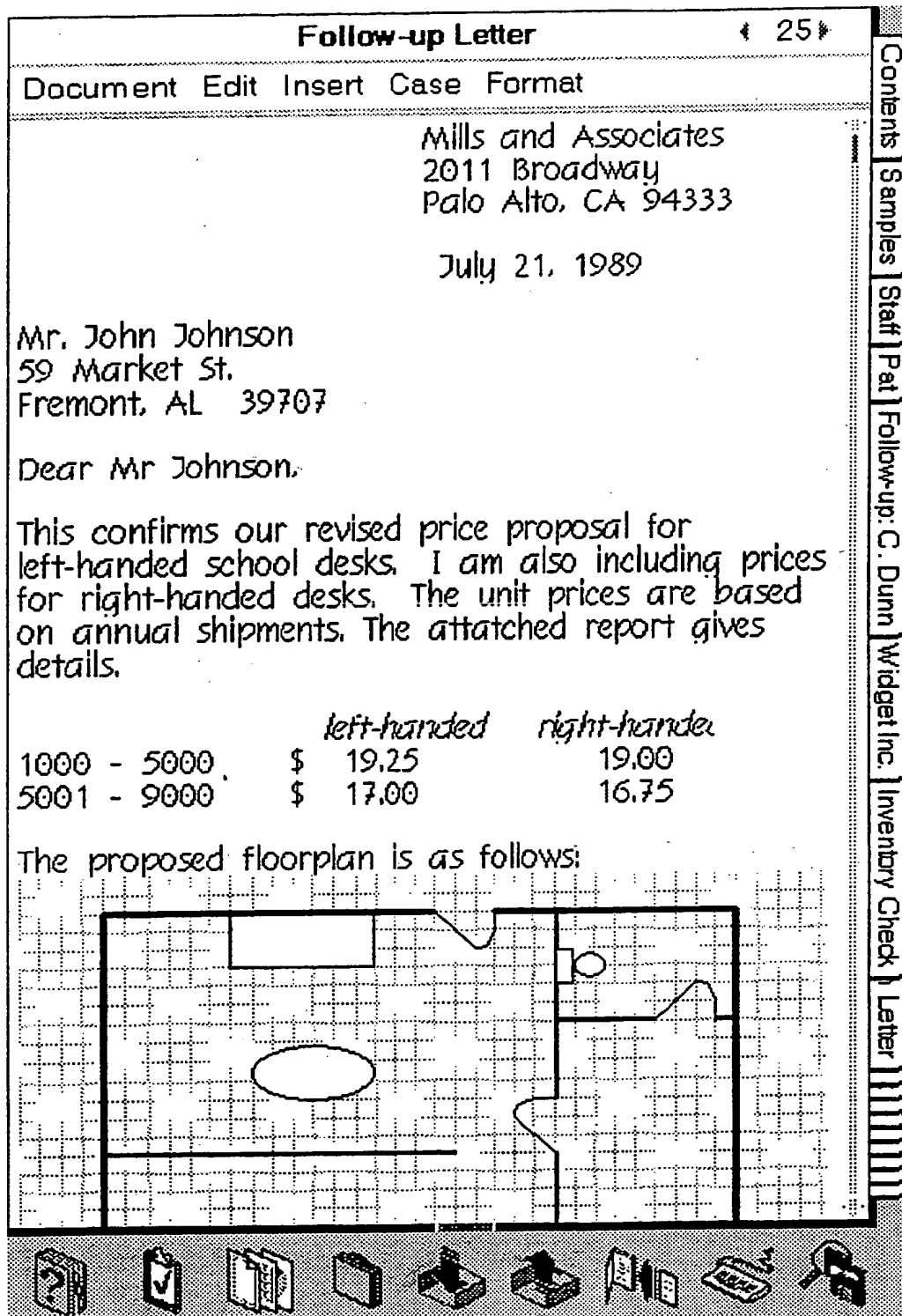


Fig. 48

INTERNATIONAL SEARCH REPORT

International Application No

PCT/US 91/08077

I. CLASSIFICATION OF SUBJECT MATTER (if several classification symbols apply, indicate all) ⁶		
According to International Patent Classification (IPC) or to both National Classification and IPC		
Int.Cl. 5 G06F15/40;	G06F9/44;	G06F3/033; G06F3/023
II. FIELDS SEARCHED		
Minimum Documentation Searched ⁷		
Classification System	Classification Symbols	
Int.Cl. 5	G06F	
Documentation Searched other than Minimum Documentation to the Extent that such Documents are Included in the Fields Searched ⁸		
III. DOCUMENTS CONSIDERED TO BE RELEVANT⁹		
Category ^o	Citation of Document, ¹¹ with indication, where appropriate, of the relevant passages ¹²	Relevant to Claim No. ¹³
X	EP,A,0 339 220 (HEWLETT-PACKARD COMPANY) 2. November 1989	1-5, 16-17, 39-45
A	see page 1, line 1 - page 1, line 44; claims 2-3,9; figure 52	6-15, 18-38, 46-71
X	HP WORLD November 1988, pages 24 - 27; PAUL BRADY: 'Personal productivity using A.I. techniques' see the whole document	1-5, 16-17, 39-45
<p>^o Special categories of cited documents : ¹⁰</p> <p>"A" document defining the general state of the art which is not considered to be of particular relevance</p> <p>"E" earlier document but published on or after the international filing date</p> <p>"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)</p> <p>"O" document referring to an oral disclosure, use, exhibition or other means</p> <p>"P" document published prior to the international filing date but later than the priority date claimed</p> <p>"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention</p> <p>"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step</p> <p>"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art.</p> <p>"A" document member of the same patent family</p>		
IV. CERTIFICATION		
Date of the Actual Completion of the International Search	Date of Mailing of this International Search Report	
10 MARCH 1992	26. 03. 92	
International Searching Authority	Signature of Authorized Officer	
EUROPEAN PATENT OFFICE	KATERBAU R.E. <i>R. Kerban</i>	

III. DOCUMENTS CONSIDERED TO BE RELEVANT (CONTINUED FROM THE SECOND SHEET)

Category ^o	Citation of Document, with indication, where appropriate, of the relevant passages	Relevant to Claim No.
A	HP WORLD October 1988, pages 11 - 12; PAUL BRADY: 'New waves of personal productivity' see the whole document ---	1-4, 46, 55, 68
A	ACM TRANSACTIONS ON GRAPHICS vol. 5, no. 2, April 1986, pages 79 - 109; R.W.SCHEIFLER ET AL.: 'The X Window System' cited in the application see page 82, paragraph 4 see page 83, paragraph 7 see page 89, paragraph 4 ---	1-4, 46, 55, 68

THIS PAGE BLANK (USPTO)